

# PROGRAM BITEN 79004

GOD HELG  
GOD HELG



## INNEHÅLL

Dyre Medlem	s 2
Medlemsinbetalning	s 3
KALLELSE TILL ÅRSMÖTET	s 4
Box 1	s 5
Tips (Blio)	s 6
Systemtips	s 10
Bli bekant med PC100/TI-59	s 12
X <sup>3</sup> =i	s 14
Novisen vid TI-59:an	s 18
Digital clock	s 20
Electronic Enginee.paq.	s 21
Praktiska tips om snabbare..	s 22
Hexy	s 23
Programredovisningsblanketter	s 24
Något om TI-59s datorliknande	s 26
Smådatormarknaden	s 29
Screentryck	s 30
Litteratur	s 36
Sortering av tider med tillh text	s 38
Pgm-bitar	s 40
Datalagring-testning-statistik	s 48
Roulett med TI-59	s 57
Mall	s 61



Föreningen för Utveckling av Program till Räkna

KALLELSE  
till  
ÅRSMÖTE

Lokal: Gunnar Svanborgs gillestuga på adress Djupdalsvägar 127, 161 44 BROMMA  
Dagordning:

1. Mötets Öppnande. Fråga om mötets behöriga utlysande. Ev godkännande av dagordning.
2. Val av 1) Ordförande för mötet  
II) Sekreterare för mötet
3. Val av två justeringsmän att jämte ordföranden justera protokollet.
4. Styrelsens berättelse om verksamheten.
5. Revisorns berättelse.
6. Fråga om ansvarsfrihet för styrelsen.
7. Val av a) styrelse  
b) PROGRAMBITENS tidsningsredaktion  
c) valberedning  
d) revisorer  
e) övriga ombud
8. Behandling av årsplan verksamhetsåret 1980.
9. Till styrelsen inkomna förslag.
10. Övriga frågor.
11. Fastställande av tidpunkt för nästa möte.
12. Mötets avslutande.

Förslag som önskas behandlat på årsmötet skall vara styrelsen (sekreteraren) till handa senast den 4:e februari 1980.

Anmälan om deltagande bör vara skriftligen till Box 1, 740 30 BJÖRKLINGE innan den 14:e februari 1980. Ändring av lokal kan då meddelas (om vi blir för många) via brev eller direkt på ingången. Försök därför att meddela ankomst i god tid före mötet. Lokalen öppnar en halvtimme innan fastställt klockslag, för den som önskar läsa protokoll och kassabok.

Väl mött!

  
Ordföraren

Jag anmäler mitt deltagande i ÅRSMÖTET den 23 februari 1980 kl 1500 med att insända betalning för makor, 01 (te) mm på föreningens eller annat postgiroinbetalningskort uppgående till s:a 20,-, samt angivande på den rutade sidan att betalningen gäller närvaro vid årsmötet.

PROGRAMBITEN 79-3/4

Red.

## Box 1 740 30 BJÖRKLINGE

!

Kom ihåg att märka brevet med "red" om åsikten ska få inflyande!

Jag beställer följande ur programbiblioteket:  
TIPSY B3ic  
SORTERING AV TIDER B3ia  
MÅNADSKALENDER B6k

Obs! Beställningar görs endast genom inbetalning via postgiro!  
Föreningen måste minska på administrativa åtaganden p g a tidsskäl.

Tyvärr har kodbeteckningarna blandats ihop i en del sammanhang.  
I PROGRAMBITEN 78-2: sid 29  
Sateellitpörning, Prognos-tips, Pokerspel, skall ges kod till i stället för P1111  
P11a på samma sida och den föregående skall bytas mot P11b.  
Flygnavigering äges av P2hf. P1a

Betr. egenvärden till en matris, som en medlem frågade om ett halvår sedan. Ett kort program att ta fram dessa har kommit föreningen till kännedom.

## VARNING!

Efter att ha använt gnusgbokstöver för skrift på magnetkort en längre tid har det visat sig att bokstöverna lossnar om korten förvaras i födräl.

Jag vill därför varna för att använda gnusgbokstöver. P1a

Med anledning av ledaren i PROGRAMBITEN 78-2 där det står att medlemsår skall sammanfalla med kalenderår: Betyder det att jag skall betala in ny medlemsavgift i december för år 1980? (JA)  
I samma ledare står det att ni ska ge ut 6 nr på medlemsavgiften. Kommer då de två numren som gavs ut 1978 att medräknas i dessa 6 nummer? (JA)  
Eller kommer jag att få 6 nummer från 1979 och de två numren från 1978 var "frinummer"? (NEJ)  
Detta alternativ tycker jag är det rimligaste, för vem prenumerar på en tidsskrift och erhåller flera nummer bakåt som räknas in i prenumerationen? Q11o

Svar:  
Observera att PROGRAMBITEN är ett informationsorgan för medlemmar i Föreningen för utveckling av program till räkna (det är alltså inte en prenumeration) vilken utlovades i sex exemplar på medlemsavgiften. Detta har nu uppfyllts. Många tillkommande medlemmar har önskat PROGRAMBITEN från uppstarten, och tillkommande medlemmar skall naturligtvis ha rätt till allt material som utkommer under medlemsåret.

Årsmötet får ta upp frågan om "löpande medlemsår".

Varför har inte tidningen börjat recensera programmoduler och programböcker?

De återförsäljare som "säljer" TI-produkter tar bara hem moduler och programböcker på direkt beställning. Den enda service man får är att de uppger vad man får betala. I övrigt är det som att köpa grisen i skäken.

För några månader sedan lät jag beställa hem "Printer Utility Packette". Pris 60 kr. Vad fick jag för pengarna? Svar: En radda fullkomligt värdelösa program.

1. Vem har nytta av en "Printing Clock"?
2. Vem har nytta av en "Cartesian Graf"?

Med dålig noggrannhet plottar det ut punkter i talplanet. Running time: 5 minuter.  
Det går fortare och bättre att göra det för hand.

3. Samma för "Function plotter".
4. Samma för "Bar graf plotter".
5. Samma för "Memo pad".

Det sistnämnda programmet ger möjlighet att använda skrivaren som någon slags skrivmaskin. Vem kan ha någon användning av det?  
En penna och en pappersbit fungerar lika bra, och är dessutom billigare.

Då återstår bara "Flagtester", "Alfa-numeric register listing" och "TI-59 Banner program".  
Det sistnämnda måste ändras för att få med svenska bokstöver och de två förstnämnda är inte svårare än att man kan skriva dem själv, om man har behov av dem.

Hälsningar  
Sven Utberg  
P1p

Tack för recensionen. Välkommen med flera. Red.

PROGRAMBITEN 79-3/4





15-16 december 1979

1 Bolton-Spetch  
2 Brighton-Stoke  
3 Coventry-Manch 1  
4 Leeds-Wolverhampton  
5 Liverpool-Crystal P  
6 Tottenham-Aston Villa  
7 West Bromwich-Arsenal  
8 Charlton-Leicester  
9 Newcastle-Queens PR  
10 Oxford-Hull City  
11 Shrewsbury-West Ham  
12 Watford-Sunderland  
13 Wrexham-Luton

**SYSTEM TIPS**

**HUR MÅNGA RADER  
12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1**

Varje rader gäller  
Vissa rader gäller  
Vissa rader gäller

Som måndagsdag  
**TORSDAG 18.12.79**

Varje  
När  
På  
På  
Örn

**50**

Ja, så lätt är det att få 13, 12, 11 eller 10 rätt - när man vet hur det har gått. Dessförinnan är det förstås svårare.

Värdet med detta program är att det gör det lättare att kontrollera att man blivit rätt noterat i vinst-listan. I annat fall bör man som bekant protestera i tid. Se tipstalongens bakside. Lycka till!

**Exempel 1.**

I ett system med 4 halvgarderingar + 3 helgarderingar (= 432 rader) har du haft tur och fått 0 fel i både ogardeade och gardeade. Resultatet blir att du har:

- 1 rad med 13 rätt
- 10 rader med 12 rätt
- 42 rader med 11 rätt
- 96 rader med 10 rätt

samt dessutom 9:or, 8:or, 7:or och 6:or till ett sammanlagt antal av 283 st.

Detta är det bästa som kan hända dig med ett 432-raders system. (EX.3 är det näst bästa, se nedan.)

**Exempel 2.**

I samma system som i exempel 1 har du fått ett fel i ogardeade och 0 fel i gardeade.

Här ger systemet:

- 1 rad med 13 rätt
- 10 rader med 11 rätt
- 42 rader med 10 rätt

samt 379 st med 9 rätt eller mindre.

Som du ser - och som man kunde vänta sig - så blir hela resultatserien bara nedflyttad ett steg från 13 / 6 r till 12 / 5 r. - Ska man nu ha ett fel, så ska det tydligen inte ligga i en ogardead. (Jmr Ex.3).

**Exempel 3.**

Återigen samma system som i exempel 1 och 2, men nu med 0 fel i ogardeade och 1 fel i gardeade.

Detta ger:

- 2 rader med 12 rätt
- 18 rader med 11 rätt
- 66 rader med 10 rätt

samt 346 st med 9 rätt eller sämre.  
Alltså nästan dubbelt så bra som Ex.2, men förstås inte alls så bra som Ex.1.

**Exempel 1.**

P23A/143	HALVGÅRD=?	HELGÅRD=?	SYSTEM, RADER
4.	4.	3.	432.

FEL I OGDAR.? 0.	FEL I GÅRD.? 0.
13.	12.
19.	11.
12.	7.
11.	10.
18.	10.
20.	20.
9.	8.
8.	8.

**Exempel 2.**

P23A/143	HALVGÅRD=?	HELGÅRD=?	SYSTEM, RADER
4.	4.	3.	432.

FEL I OGDAR.? 0.	FEL I GÅRD.? 0.
11.	12.
12.	2.
11.	11.
7.	14.
10.	10.
18.	36.
9.	40.
8.	8.
8.	16.

**Exempel 3.**

P23A/143	HALVGÅRD=?	HELGÅRD=?	SYSTEM, RADER
4.	4.	3.	432.

FEL I OGDAR.? 0.	FEL I GÅRD.? 1.
12.	12.
11.	11.
14.	14.
10.	10.
36.	36.
9.	9.
40.	40.
8.	8.
16.	16.

307	61	CTD
308	02	02
309	28	98
310	43	RCL
311	03	03
312	85	+
313	02	+
314	28	3
315	95	+
316	42	STO
317	11	11
318	43	RCL
319	06	06
320	67	EQ
321	03	03
322	30	30
323	44	SUM
324	04	04
325	69	DP
326	26	26
327	61	CTD
328	03	03
329	18	18
330	43	RCL
331	04	04
332	85	+
333	02	+
334	03	3
335	95	+
336	42	STO
337	10	10
338	02	2
339	02	2
340	42	STO
341	12	12
342	02	2
343	45	YX
344	43	RCL
345	09	09
346	65	+
347	73	RC*
348	10	10
349	10	10
350	43	RCL
351	04	04
352	85	+
353	02	+
354	03	3
355	95	+
356	42	STO
357	10	10
358	02	2
359	02	2
360	42	STO
361	12	12
362	02	2
363	43	RCL
364	22	22
365	06	06
366	06	06
367	43	RCL
368	03	03
369	42	42
370	05	05
371	13	13
372	75	+
373	01	1
374	95	+
375	43	RCL
376	13	13
377	42	STO
378	01	1
379	01	1
380	44	SUM
381	11	11
382	22	22
383	97	ISZ
384	10	10
385	00	00
386	03	03
387	03	03
388	03	03
389	93	93
390	61	CTD
391	03	03
392	38	38
393	01	1
394	04	4

Detta är inte ett program som ger råd och anvisningar om hur du skall tippa. Det finns det andra program som gör (se t.ex. Fbreningsens programbibliotek). Det här programmet talar om i efterhand - när du vet hur du har tippat, och hur det verkligen gick - hur många rader med 11 rätt, 12 rätt och 13 rätt som du fick på ditt system.

Du behöver veta 4 saker när du skall använda programmet:

1. Hur många halvgarderingar innehåll systemet?
2. Hur många helgarderingar?
3. Hur många av de ogardeade (säkra) blev fel?
4. Hur många av de halvgardeade blev fel? (Av de halvgardeade kan ju inga bli fel!)

**Programmet har en begränsning**

Antalet garderingar (halv + hel) får ej överstiga 8. Som bekant har AB Tipstjänst dessutom den begränsningen att antalet rader i ett system ej får överstiga 486, dvs max 5 helgarderingar + 1 halvgardering.

**BRUKSANVISNING**

1. Uppdelning = standard = 479,59
2. Programmera enligt listan.
3. Spela in programmet på 2 kortsidor om du vill spara det.
4. Tryck A
5. Besvara de fyra frågorna i tur och ordning enligt de fyra punkterna ovan. Tryck R/S efter varje svar.
6. Invänta utskrift. Det kan i vissa fall ta över en minut från första R/S. Betydelsen av utskriften framgår av exemplen.



# Bli bekant med PC100+TI-59 och lämför med basic

Med anledning av en artikel i Radio&Television om grunderna i BASIC vill jag försöka mig på att överföra den praktiska introduktionen till lämpat på TI-58/59 med skrivare.

För att utvärdera vari skillnaden mellan BASIC och TI-språket ligger kan vi försöka åstadkomma typ av samarbete med TI-58/59-an som utvecklades i artikeln. Exemplet gällde en multiplicering av 5x6. Att göra beräkningen är ju ytterst enkelt på vår räknare, därför har för vi programmeringen närmast det gäller att fixa till en utskrift av resultatet.

Först kan vi simulera den allra enklaste interaktiva formen av samarbete genom att trycka ned Trace-tangenten eller med ST flg 9. Då får vi utskrift efter varje åtgärd, ungefär som det fungerar för Basic-användaren varje gång han skriver ordet PRINT. Skillnaden ligger främst här i att vi slipper stava till ordet PRINT varje gång vi vill få ett svar nedskrivet. På köpet får vi dessutom utskrift för varje åtgärd.



Σ ← 2

Antag nu att vi gör exemplet att beräkna priset för 5 styck av en vara å 6 kronor och önskar att lägga moms på summan. Hur gör vi då?

Ja, om vi önskar göra en sådan beräkning mer än två gånger är det kanske värt att göra ett kort program omkring detta. Innan vi instruerar räknaren att vi vill programmera är det dock säkrast att trycka RCT (Reset=nollställ) vilket gör att vi säkert befinner oss på programsteg 000 när vi med tryckningen LRN (Learn=lära) "övergår" i programmeringsläge.

LRN

Nu visar displayen 000 00, där de tre högra nollorna visar att vi befinner oss på detta programsteg och de två högra nollorna är koden för den instruktion som skall utföras på detta programsteg. Vi behöver inte bekymra oss om att programmeringen blir så rätt sådär direkt för första "programmeringsförsöket", vi har ju alltid möjlighet att rätta till en felaktig instruktion genom att "bakka tillbaka" med BST och att helt enkelt rätta till felet med ATT skriva "över det gamla" eller, om det behövs, ge uttrycket för en extra instruktion med Ins (=Insert=Stoppa in).

För att slippa alltför krångliga tryckningar för att kunna använda programmet i ett senare läge börjar vi med att ge programmet ett "namn" som är lättåtkomligt: LBL A i början av programmet innebär att man kan starta programmet med tangenten A. Vi börjar alltså med att trycka 2nd LBL A (i fortsättningen utelämnas 2nd i texten).

I det enklaste fallet kan vi då tänka oss att låta programmet sköta momsbekräningen med tangenten Trace på skrivaren fortfarande nedtryckt. Vi måste så tänka oss in i var i beräkningen vi vill trycka på "Moms-tangenten", det gäller nämligen att låta programmet "överta" så mycket som möjligt av beräkningen. Det allra minsta som är nödvändigt är göra innan man trycker på A blir 5x6, resten av den nödvändiga trycksekvensen kan alltså "läggas" in i A (X L.2065 z). Programmet avslutas så med INV SBR eller R/S (Run/Stop) Programmet får då utseendet så som visas nedan till vänster och utskriften blir som till höger.

```

000 76 LBL
001 11 A
002 65 X CLR
003 01 1 5. X
004 93 1.2065 X
005 02 2 36.20 X
006 00 0 R/S
007 06 6
008 05 5
009 95 =
010 91 R/S

```

```

1 INPUT A
2 INPUT P
3 X←ATP
4 PRINT X
5 PRINT L.2063
6 END

```

När vi skriver RUN och verkställer får vi följande resultat

och hela svarskriften, med de inmalade värdena

```

30
36.189
samt dessutom END AT LINE 6
och i utprepar in te det i följande.

```

Med 13 tryckningar åstadkom vi alltså ett program med enklaste utskrift, att lämföra med ett Basic-program som skulle kräva 48 nedtryckningar + tre tryckningar för att starta programmet (RUN)

Men såhär kan man naturligtvis inte lämna ett program!

För att andra skall ha nytta av det, och du själv när du glömt hur det vara avsett att användas, måste du göra det "instruktionsstyrt"! Här kommer din chans; aldrig tidigare har människan haft ett sådant tillfälle till att formulera bra svar, här ligger en av tjušnjargarna med att programmera. Det gäller ju att avlasta honom/henne från så mycket tankearbete som möjligt.

Hur är då detta möjligt med vårt program?

Vi försöker att det är en försäljare, som ej äger tid att sätta sig in i programmet, skall använda det.

Det första vi då gör är att ta bort den automatiska utskriften (frigör Trace-tangenten eller trycker INV ST flg 9). Nu får vi lov att sköta utskriften med Print och Q-koderna.

Om vi till exempel sätter in Print efter att 5x6 uträknats (efter X-tecknet på rad 002; stega 2 till rad 003 med ST resp BST och tryck Inv Print) samt efter monsen har beräknats (stega fram med SET till 010 och tryck Print R/S), erhålles en utskrift av ungefär samma kvalitet som tidigare.

För att nu åstadkomma skriftväxling mellan användare och räknare får vi ett mera lidande arbete än om vi hade haft en hemdators tangentbord; det gäller nämligen att mata in sifferkodder för bokstäver i programmet samt att fixa till en utskrift av dessa med hjälp av Op 01-05.



Först gäller det dock att bestämma sig för vad den första tryckningen på A skall stå för. Det finns framför allt två möjligheter; Antingen skriver vi START ovanför A och låter programmet vara helt instruktionsstyrt, eller också skriver vi ANTAL ST ovanför A och låter användaren själv tänka ut att antalet av varan skall vara det första som matas in vid programmering.

Det bekvämare för användaren är naturligtvis att trycka start, men om beräkningen behöver göras flera gånger blir det tråkigt att vänta på att frågan ska skrivas ut före varje inmatning (utskriften går ju ganska långsamt).

025 65 X 055 03 3 36370062. 21  
026 99 PRT 056 95 = 3335243600. 22  
027 02 2 057 69 DP 63000363762. 23  
028 02 2 058 06 06 3327413600. 24  
029 16 A 059 91 R/S 30323036. 25

START	ANTAL ST	PRIS/ST
-------	----------	---------

För att efterlikna exemplet i R&T skriver vi START ovanför A, men för att samtidigt göra det rationellt att använda skriver vi ANTAL ovanför B-tangenten! För att göra det enkelt för oss trycker vi nu END A och "går över" i programmeringsläge med LRN. Därefter matar vi in koden för första frågan, ANTAL ST; (det gäller att vara kortfattad, texten tar ju också av värt begränsade utrymme).

Här kan man göra på många sätt. Jag föredrar att lägga texten i minnen och låta utskriften ske i en subrutin vi kan kalla A' så länge.

ANTAL ST:

Vi börjar då med att lägga texten i minnen, ex minne 20 och 21. Tryck 1331371327 END, 20 36370062 END 21. Utskriftsrutinen kan vi lägga i början och kalla A'. Tryck BST LRN (vi kommer nu till "skriva över" det gamla programmet, det är tyvärr det rationellaste!).

LBL A' STO 00 RCL ind 00 Op 01 Op 20 RCL ind 00 Op 02 Op 05 INV SBR skriver ut text ur två minnen med början i det som är angivet i displayen då subrutinen "kallas".

Efter att på detta sätt ha förberett räknaren på "skrivarbete" tar vi så till med programmet: Vi befinner oss på programsteg 017 och matar in LBL A 20 MA 21 är vilket ger oss den nämnda utskriften. Räknaren stannar nu exakt här vid programkörningen, och vi kan därför fortsätta med

LBL B X Tr 22 A' R/S Fix 2 LBL C' Prt Adv = X Prt x 24 A' 2635 Op 04 1.2065 = Op 06 R/S.

000 76 LBL	030 91 R/S	ANTAL ST :
001 16 A'	031 98 Fix	PRIS / ST :
002 42 STD	032 02 02	6.00
003 00 00	033 76 LBL	
004 73 RC=	034 13 C	
005 00 00	035 99 PRT	
006 69 DP	036 98 ADV	30.00
007 01 01	037 95 =	PLUS MDMS
008 69 DP	038 85 X	36.19
009 20 20	039 99 PRT	KR
010 73 RC=	040 32 X:IT	
011 00 00	041 02 2	
012 69 DP	042 04 04	
013 02 02	043 16 A'	
014 69 DP	044 02 2	
015 05 05	045 06 6	
016 92 RTN	046 03 3	
017 76 LBL	047 05 5	
018 11 R	048 69 DP	
019 02 2	049 04 04	
020 00 0	050 01 1	
021 16 A'	051 93 .	
022 91 R/S	052 02 2	
023 76 LBL	053 00 0	
024 12 B	054 06 6	1331371327. 20
025 65 X	055 03 3	36370062. 21
026 99 PRT	056 95 =	3335243600. 22
027 02 2	057 69 DP	63000363762. 23
028 02 2	058 06 06	3327413600. 24
029 16 A'	059 91 R/S	30323036. 25



```

5 PRINT "ANTAL
VAROR:"
10 INPUT A
15 PRINT "PRIS PER
STYCK (KR):"
20 PRINT P
30 X←AP
35 PRINT "SUMMA UTAN
MOMS:"X:KR
50 PRINT "SUMMA MED
MOMS:"X*L.2063:"KR"
60 END

```

```

ANTAL VAROR:
PRIS PER STYCK (KR):6
SUMMA UTAN MOMS
3 KR
SUMMA MED MOMS
36.189 KR

```



Hela programmet tog 84 tangenttryckningar + att 10x6=60 siffror skall läggas in i minnen 20-25 enligt följande: 3335243600 STO 20 - 6300036762 STO 23 3327413600 STO 24 - 30323036 STO 25. Sammanlagt 144 tangenttryckningar mot 169 för en något "värtaligare" basic-variant.

Totala antalet nedtryckningar är alltså jämförbara för Hemdatorn med basic resp TI-58/59, men med basic blir datorn något värtaligare. Kodsystemet är naturligtvis till nackdel för TI-59 när man vill att sig koderna går det inte alltför långsamt.

[111]

$$X^3 = i$$

$$?$$

**BRUKSANVISNING utan skrivare**

1. Gör 2 OP 17 och "Läs" i programmet
  2. Tryck A Invänta display = 1 Mata in real-delen av B
  3. " " " " " " " " 2 " " imag- " " B
  4. " " " " " " " " 3 " " real- " " C
  5. " " " " " " " " 4 " " imag- " " C
  6. " " " " " " " " 5 " " real- " " D
  7. " " " " " " " " 6 " " imag- " " D
  8. Tryck R/S
- Efter ca 10 min.: Display = real-delen av  $x_1$
9. Tryck R/S " " imag- " "  $x_1$
  10. " " " " " " " " 1. " " real- " "  $x_2$
  11. " " " " " " " " " " " " imag- " "  $x_2$
  12. " " " " " " " " " " " " real- " "  $x_3$
  13. " " " " " " " " " " " " imag- " "  $x_3$

**BRUKSANVISNING med skrivare**

1. Gör 2 OP 17 och "Läs" in programmet
2. Tryck A
3. Besvara frågorna, som skrivs på remsan

Efter ca 10 min. skrivs automatiskt och med sina beteckningar real- och imag-del för de tre rötterna.

**Exempel:**

$$x^3 + (1+2i)x^2 + (3+4i)x + (5+6i) = 0$$

A = +1

REALDEL AV B = ?

IMAGDEL AV B = ?

REALDEL AV C = ?

IMAGDEL AV C = ?

REALDEL AV D = ?

IMAGDEL AV D = ?

REALDEL AV E = ?

IMAGDEL AV E = ?

REALDEL AV F = ?

IMAGDEL AV F = ?

REALDEL AV G = ?

IMAGDEL AV G = ?

REALDEL AV H = ?

IMAGDEL AV H = ?

REALDEL AV I = ?

IMAGDEL AV I = ?

REALDEL AV J = ?

IMAGDEL AV J = ?

REALDEL AV K = ?

IMAGDEL AV K = ?

REALDEL AV L = ?

IMAGDEL AV L = ?

REALDEL AV M = ?

IMAGDEL AV M = ?

**LÖSNING AV 3:e gradsekvationer med KOMPLEXA KOPFFICIENTER**

Vilka rötter har ekvationerna  $x^3 = 1$  och  $x^3 + x^2 + x = -1$   
Dessa och liknande frågor kan du få svar på med hjälp av följande program för TI-59.

Detta program behandlar i ekvationen:

$$Ax^3 + Bx^2 + Cx + D = 0 \quad (1)$$

det specialfallet att A = +1 d v s fallet:

$$x^3 + Bx^2 + Cx + D = 0 \quad (2)$$

och med B, C och D som komplexa tal.

Man kan givetvis - genom division med A - överföra alla 3:e gradsekvationer med komplexa koefficienter till denna form - även sådana där också A är ett komplext tal.

I ekv (2) gäller:

$$p = C - \frac{1}{3}B^2$$

$$q = D - \frac{1}{3}BC + \frac{2}{27}B^3$$

$$\Delta = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{27}\right)^3 \left(\frac{q}{2}\right)^2$$

$$u_1 = \sqrt[3]{\frac{p}{3\Delta} - \frac{q}{2}}$$

$$u_{2,3} = u_1 \left(-\frac{1}{2} \pm \frac{\sqrt{3}}{2}i\right)$$

$$v_n = -\frac{p}{3u_n}$$

$$r_n = u_n + v_n - \frac{B}{3}$$

och på dessa samband är programmet uppbyggt.

Programmet, som är mycket litetskött, arbetar med utdelningen 799.19 (2 OP 17), uppstår stegen 000-772, använder samtliga register (R00-R19) samt t-registret och modulprogrammen M104 och M105 och har tre labels - A, A' och B' - varav de två sistnämnda endast är SER-adresser. P.ö. sker all adressering NUMERISKT, varför stor försiktighet måste iakttagas vid eventuella ändringar i pgm.

Programmet arbetar lika bra med som utan skrivare (PC100), men om det skall användas med skrivare bör 3/G uttrycks mot NOP i stegen nr 656, 673, 689, 704, 720 och 735.

**OBSERVERA** att med den metod, som används i detta program, utdrikes helt CASUS IRREDUCIBILIS genom att alla rötter - även de rent reella beräknas som komplexa. För reella rötter blir då imag-delen = 0.

P g a maskinens skyddsseffror kan noll ibland komma fram som t ex: -2. 10<sup>-12</sup>

**Exempel:**  $x^3 = +1$

A = +1

REALDEL AV B = ?

IMAGDEL AV B = ?

REALDEL AV C = ?

IMAGDEL AV C = ?

REALDEL AV D = ?

IMAGDEL AV D = ?

**Exempel:**  $x^3 = +1$

A = +1

REALDEL AV B = ?

IMAGDEL AV B = ?

REALDEL AV C = ?

IMAGDEL AV C = ?

REALDEL AV D = ?

IMAGDEL AV D = ?

**Exempel:**  $x^3 = -1$

A = +1

REALDEL AV B = ?

IMAGDEL AV B = ?

REALDEL AV C = ?

IMAGDEL AV C = ?

REALDEL AV D = ?

IMAGDEL AV D = ?

**LISTNING FÖR 1**

285	03	3	285	03	3
286	35	1/X	286	35	1/X
287	16	A'	287	16	A'
288	00	0	288	00	0
289	16	A'	289	16	A'
290	36	PGM	290	36	PGM
291	04	04	291	04	04
292	13	C	292	13	C
293	43	RCL	293	43	RCL
294	07	07	294	07	07
295	16	A'	295	16	A'
296	43	RCL	296	43	RCL
297	08	08	297	08	08
298	16	A'	298	16	A'
299	36	PGM	299	36	PGM
300	04	04	300	04	04
301	10	E'	301	10	E'
302	36	PGM	302	36	PGM
303	04	04	303	04	04
304	17	B'	304	17	B'
305	42	STD	305	42	STD
306	11	11	306	11	11
307	32	X/T	307	32	X/T
308	42	STD	308	42	STD
309	12	12	309	12	12
310	03	3	310	03	3
311	17	B'	311	17	B'
312	00	0	312	00	0
313	17	B'	313	17	B'
314	43	RCL	314	43	RCL
315	05	05	315	05	05
316	16	A'	316	16	A'
317	43	RCL	317	43	RCL
318	16	A'	318	16	A'
319	16	A'	319	16	A'
320	36	PGM	320	36	PGM
321	04	04	321	04	04
322	42	STD	322	42	STD
323	42	STD	323	42	STD
324	01	01	324	01	01
325	43	RCL	325	43	RCL
326	42	STD	326	42	STD
327	02	02	327	02	02
328	53	3	328	53	3
329	16	A'	329	16	A'
330	55	+	330	55	+
331	02	2	331	02	2
332	07	7	332	07	7
333	54	0	333	54	0
334	16	A'	334	16	A'
335	00	0	335	00	0
336	16	A'	336	16	A'
337	36	PGM	337	36	PGM
338	04	04	338	04	04
339	13	C	339	13	C
340	42	STD	340	42	STD
341	13	13	341	13	13
342	32	X/T	342	32	X/T
343	42	STD	343	42	STD
344	14	14	344	14	14
345	43	RCL	345	43	RCL
346	05	05	346	05	05
347	17	B'	347	17	B'
348	43	RCL	348	43	RCL
349	06	06	349	06	06
350	16	A'	350	16	A'
351	43	RCL	351	43	RCL
352	07	07	352	07	07
353	16	A'	353	16	A'
354	43	RCL	354	43	RCL
355	08	08	355	08	08
356	16	A'	356	16	A'
357	36	PGM	357	36	PGM
358	17	B'	358	17	B'
359	13	C	359	13	C
360	03	3	360	03	3
361	36	PGM	361	36	PGM
362	16	A'	362	16	A'
363	00	0	363	00	0
364	36	PGM	364	36	PGM
365	13	C	365	13	C
366	43	RCL	366	43	RCL
367	09	09	367	09	09

**Forts. Programlisting SYSTEMS**

```

395 75 - 434 99 PRT
396 43 RCL 435 98 ADV
397 08 08 436 61 GTO
398 75 - 437 04
399 43 RCL 438 08 08
400 09 09 439 98 ADV
401 95 = 440 98 ADV
402 42 STD 441 98 ADV
403 00 00 442 98 ADV
404 01 1 443 91 R/S
405 03 3 444 98 ADV
406 42 STD 445 36 PGM
407 12 12 446 69 DP
408 01 1 447 00 00
409 44 SUM 448 03 3
410 12 12 449 03 3
411 43 RCL 450 00 0
412 12 12 451 03 3
413 75 12 452 00 0
414 02 2 453 04 4
415 03 3 454 69 DP
416 95 = 455 01 01
417 67 EG 456 01 01
418 04 04 457 03 3
419 39 39 458 06 6
420 73 RCL 459 06 6
421 12 12 460 00 0
422 67 EG 461 02 2
423 04 04 462 00 0
424 08 08 463 00 0
425 01 1 464 00 0
426 22 INV 465 04 4
427 44 SUM 466 59 DP
428 00 00 467 02 02
429 43 RCL 468 69 DP
430 00 00 469 05 05
431 99 FTX 470 98 ADV
432 73 FC* 471 92 RTH
433 12 12
    
```

**Forts. Programlisting X<sup>3</sup> = j**

```

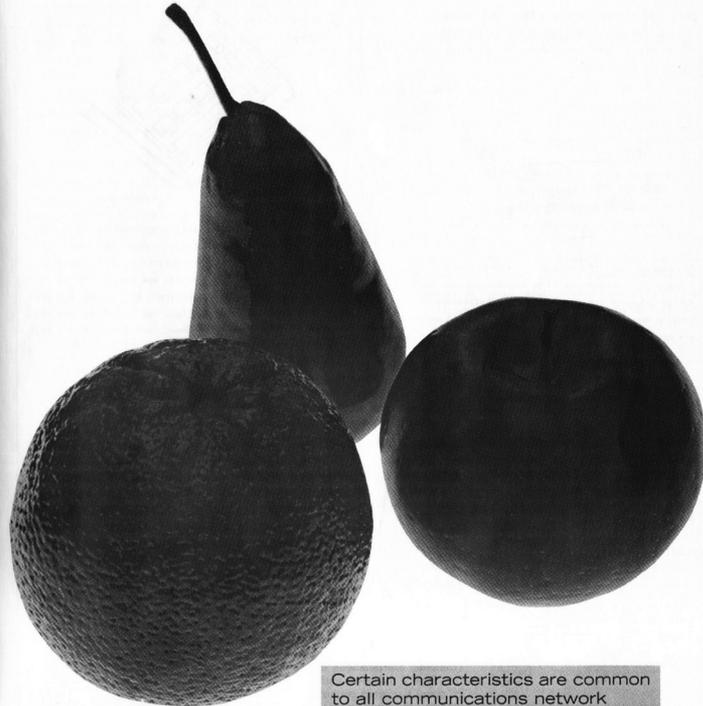
370 16 A* 415 43 RCL 460 43 RCL
371 43 RCL 416 13 13 461 08 08
372 10 10 417 55 + 462 16 A*
373 16 A* 418 02 2 463 36 PGM
374 36 PGM 419 54 4 464 04 04
375 04 04 420 17 8 465 36 PGM
376 25 05 421 53 3 466 12 B
377 36 PGM 422 43 RCL 467 05 05
378 04 04 423 14 14 468 14 D
379 17 B* 424 55 2 469 53 3
380 43 RCL 425 02 2 470 43 RCL
381 13 13 426 54 4 471 13 13
382 16 A* 427 17 B* 472 05 05
383 43 RCL 428 36 PGM 473 02 2
384 14 14 429 05 05 474 54 4
385 16 A* 430 13 C 475 16 A*
386 36 PGM 431 42 STD 476 53 3
387 04 04 432 07 07 477 43 RCL
388 12 B 433 32 XIT 478 14 14
389 42 STD 434 42 STD 479 55 + 4
390 13 13 435 08 08 480 00 00
391 32 XIT 436 03 3 481 54 4
392 42 STD 437 17 B* 482 16 A*
393 14 14 438 00 0 483 36 PGM
394 43 RCL 438 17 B* 484 04 04
395 11 11 440 53 3 485 17 B*
396 17 B* 441 43 RCL 486 03 3
397 42 RCL 442 11 11 487 16 A*
398 12 12 443 03 3 488 00 0
399 17 B* 444 03 3 489 16 A*
400 36 PGM 445 54 4 490 36 PGM
401 05 05 446 16 A* 491 01 01
402 12 B 447 16 A* 492 10 E
403 58 FTX 448 43 RCL 493 36 PGM
404 08 08 449 12 12 494 04 04
405 52 EE 450 03 3 495 15 E
406 22 INV 451 03 3 496 42 STD
407 52 EE 452 54 4 497 07 07
408 22 INV 453 16 A* 498 32 XIT
409 58 FTX 454 36 PGM 499 42 STD
410 29 CP 455 04 04 500 08 08
411 67 EG 456 14 D 501 03 3
412 07 07 457 43 RCL 502 94 +-
413 45 45 458 07 07 503 49 PRD
414 53 3 459 16 A* 504 01 01
    
```

```

505 49 PRD
506 02 02
507 43 RCL
508 11 11
509 16 A*
510 43 RCL
511 12 12
512 16 A*
513 36 PGM
514 04 04
515 16 A*
516 36 PGM
517 04 04
518 18 C
519 42 STD
520 13 13
521 32 XIT
522 42 STD
523 14 14
524 03 3
525 94 +-
526 22 INV
527 49 PRD
528 05 05
529 22 INV
530 49 PRD
531 06 06
532 17 B*
533 01 01
534 03 03
535 94 +-
536 16 A*
537 36 PGM
538 04 04
539 13 C
540 42 STD
541 09 09
542 32 XIT
543 42 STD
544 10 10
545 71 SBR
546 01 01
547 20 20
548 42 STD
549 15 15
550 32 XIT
551 42 STD
552 16 16
553 71 SBR
554 01 01
555 03 03
556 16 A*
557 36 PGM
558 04 04
559 00 00
560 42 STD
561 00 00
562 32 XIT
563 42 STD
564 19 19
565 01 SBR
566 20 20
567 42 STD
568 17 17
569 32 XIT
570 32 XIT
571 42 STD
572 18 18
573 03 03
574 94 +-
575 22 INV
576 49 PRD
577 00 00
578 22 INV
579 49 PRD
580 19 19
581 22 INV
582 49 PRD
583 09 09
584 22 INV
585 49 PRD
586 10 10
587 43 RCL
588 07 07
589 17 B*
590 43 RCL
591 08 08
    
```

```

685 43 RCL
686 15 15
687 69 DP
688 06 06
689 91 R/S
690 06 06
691 04 4
692 04 4
693 04 4
694 00 0
695 03 2
696 02 2
697 04 4
698 69 DP
699 04 04
700 43 RCL
701 16 16
702 69 DP
703 06 06
704 15 15
705 98 ADV
706 06 6
707 04 4
708 04 4
709 04 4
710 00 0
711 04 4
712 03 3
713 05 5
714 69 DP
715 04 04
716 43 RCL
717 17 17
718 69 DP
719 06 06
720 91 R/S
721 06 6
722 04 4
723 04 4
724 04 4
725 00 0
726 04 4
727 02 2
728 04 4
729 69 DP
730 04 04
731 43 RCL
732 18 18
733 69 DP
734 06 06
735 91 R/S
736 98 ADV
737 98 ADV
738 98 ADV
739 98 ADV
740 98 ADV
741 25 CLR
742 91 R/S
743 61 GTO
744 11 R
745 03 3
746 17 B*
747 00 0
748 06 06
749 43 RCL
750 13 13
751 04 +-
752 16 A*
753 43 RCL
754 14 14
755 94 +-
756 16 A*
757 36 PGM
758 04 04
759 42 STD
760 42 STD
761 07 07
762 32 XIT
763 42 STD
764 08 08
765 00 0
766 42 STD
767 13 13
768 42 STD
769 14 4
770 61 GTO
771 05 05
772 24 24
    
```



Certain characteristics are common to all communications network architectures. Unfortunately, that doesn't make them directly comparable.

PHOTO: BOB REED; JOAN LESSER, /ETCETERA

## NOVISEN VID TI-59an

Under den här rubriken, som återkommer i några nummer framöver, kommer några försök att göra sbr för att underlätta i portgången för lyckliga nyblivna TI-58/59 ägare.



Råd nummer 1 är: LXS programmeringshandboken! I den finns grunderna till programmeringstekniken. Man åstadkommer trots detta inga sofistikerade program med en gång, men man undviker en del onödigt besvär redan i början.

**LRN**

Nåväl, nu har man i alla fall införskaffat ett avancerat instrument med vilket man kan åstadkomma de märkligaste resultat.

Vad då t.ex?  
Ja, redan från början kanske det är svårt att föreställa sig detta. Det är först sedan man lekt med maskinen ett tag som man börjar få en känsla för vad den kan göra, eller rättare, vad man kan beordra den att göra.

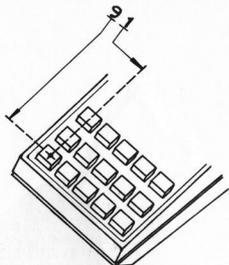
Medlemmar med introduktionstips är välkomna med bidrag.

Ins  
**SST**

Del  
**BST**

När man börjar programmera skall man vara medveten om att det första gångerna åtgör många programsteg för att maskinen skall göra något. Man bör dessutom vara beredd på att man redan från början säkerligen blir tvungen att "gå in" och ändra programmet p.g.a att det inte fungerar.

Dårför bör du på ett tidigt stadium lära dig att stega framåt SST och bakåt BST i programmet. Likaså är det Ins , som sätter in nya programsteg, och Del , som tar bort programsteg, nyttiga att kunna redan från början.



För att veta var man är i programmet är det också nyttigt att lära sig tangentkoderna och samtidigt anteckna varje tangentnedtryckning. Detta underlättar överblicken över vad som behöver rättas till. Har man tänkt skaffa skrivaren, PC-100, bör man göra detta så snart som möjligt. Med PC-100 har man möjlighet att lista programmet, och den är därför till stor hjälp i början.



För TI-58/59 ägare skall heller inte förglömmas möjligheten att få stopp på programmet där felet ligger genom att man sätter flagga 8 (St fig 8) innan programkörningen. (Observera att man på detta sätt bara hittar formella fel, inte tankefel!) Som regel bör denna flagga alltid sättas vid testkörning av program. OBS! RST nollställer flaggan.



## Vad kan då maskinen göra?

Ja, den första nyttan man har av den är om man behöver göra en beräkning om och om igen mer än tre-fyra gånger. På grund av att programmeringen kräver några extra tangentnedtryckningar och att programmeringsfel kan uppstå, kan programmering + en körning i allmänhet tidsmässigt jämföras med tre-fyra likadana beräkningar utan program. Snart nog upptäcker man att den är snabb på att köra igenom det man programmerat, och det är då man kan börja tänka sig in i räknares sätt att arbeta.

Man märker då också att det är svårt att få räkaren att göra något komplicerat, mens det däremot är lätt att få den att göra enklare beräkningar hur många gånger som helst.

Man börjar då fundera ut korta lätt programmerade sekvenser som utträttar något, vad som helst som fungerar, t.ex. en klocka.

Nedanstående enkla program visar en klocka som stannar efter en minut (60 sekunder):

000	00	00
001	42	STO
002	01	01
003	06	6
004	00	0
005	42	STO
006	00	00
007	43	RCL
008	01	01
009	66	PRN
010	01	1
011	44	SUM
012	01	01
013	68	NOP
014	68	NOP
015	68	NOP
016	68	NOP
017	68	NOP
018	68	NOP
019	68	NOP
020	68	NOP
021	68	NOP
022	68	NOP
023	68	NOP
024	97	DSZ
025	00	00
026	07	07
027	91	R/S
028	00	0

NOP-instruktionerna fördröjer pgm-behandlingen så att 1 minut erhålles. Antalet NOP-ar varierar från räkare till räkare.

DSZ-instruktionen avgränsar om 1 minut har gått eller ej.

Däremot är vi inne på en ytterligt användbar metod vid programmering - nämligen den att göra loopar.

**LOOPAR**

En beräkningsloop är konsten att göra beräkningar om och om igen, kombinerat med enkla beslut som t.ex. x=t, x=t, x=t, etc. Hela tiden prövas om ett visst värde uppnåtts eller ej, och när detta är fallet hoppar programbehandlingen ur loopen.

Loopen kan också vara färsedd med en varvräkare så att programbehandlingen hoppar ur loopen efter ett visst antal varv.

Uthopp kan alltså ske efter jämförelse med ett visst siffervärde. Detta värde "göms" i det så kallade t-registret, genom att tangenten x t trycks ned. Följande tester kan göras mot t-registret : x=t, x=t, x=t och x=t . Om t.ex. sifferföinstret visar det värde som också finns i t-registret och x=t ligger som nästa instruktion i programmet, hoppar programbehandlingen till den adress (Label- eller absolutadress) som ligger direkt efter denna instruktion.

Så länge som testvillkoret inte är uppfyllt hoppar programbehandlingen tillbaka till den adress (även här - Label- eller absolutadress) som står som "tvåa" efter testinstruktionen, och programbehandlingen fortsätter "mot nya djärva test".

Uthopp ur en loop kan också ske genom if fig instruktionen, där steget efter if fig anger vilken fallgata som testas följt av uthoppadressen i nästkommande steg. If fig instruktionen kan man lungt strunta i i början (se dock "Plottar utan skrivare", i första numret av Pgm-biten.)

**dsz**

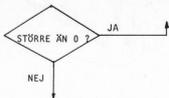
En betydligt nyttigare instruktion är varvräkaren som beordrar uthopp ur loopen efter önskat antal varv.

Den s.k. DSZ-instruktionen (Decrement and skip on zero) utför denna "trollkonst".

I vår lilla programsnutt "klockan" finns instruktionen DSZ 00 007. Instruktionen räknar ner värdet i register 00 med en enhet varje gång instruktionen genomlöps, och programbehandlingen hoppar därefter till programsteg 007. När register 00 "döms" (=0), ignoreras adressen 007 och programbehandlingen fortsätter med den efterföljande instruktionen - i detta fall R/S , varvid programmet stannar.

**HOPP OCH**

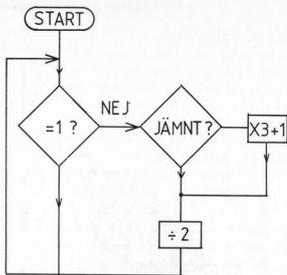
I programtexten 003 - 006 laddas register D0 med 60, vilket blir antalet varv som loopen kommer att genomgå.



Som avslutning på denna första artikel vill vi ge en liten övningsuppgift att gugga "programmeringsknöarna" över. Först ska man minimera antalet programsteg så mycket som möjligt.

Problemet lyder:

Bilda nya heltal från ett godtyckligt positivt heltal enligt regeln: Är talet jämnt, dividera med två. Är talet udda, multiplicera med 3 och addera 1. Efter ett större eller mindre antal divisioner, multiplikationer och additioner blir resultatet alltid 1. Sifferfönstret skall visa alla udda och jämna mellanresultat och programbehandlingen skall stoppa vid "1". Ett förslag till flödesschema visas härintill:



### DIGITAL CLOCK

Programmera Din dosa för realtidvisning.

Tiden presenteras i timmar, minuter och sekunder med en visning varannan sekund.

För att kalibrera klockan finns följande möjligheter:

1. Inläggning av NOP-instruktion (-er) "mellan" rad 8 och 9.
2. Ändring av den i minne 00 lagrade konstanten.

Kalibreringen är individuell för varje räknare, varför man får prova sig fram. Man bör använda instruktionen 'Fix 4', i eller utom programmet, dels för att ta bort eventuella avrundningsfel vid användning av den andra kalibreringsmetoden. Dels för att få dosen att visa alla sex siffrorna, dvs inte skära bort en nolla i antalen av sekunderna, som tillsammans med minuterna är presenterade som decimaler.

### DIGITAL CLOCK

```

000 43 RCL
001 00 00
002 44 SUM
003 01 01
004 43 RCL
005 01 01
006 22 INV
007 88 DMS
008 66 PAU
009 86 PHU
010 81 RST
011 00 0
  
```

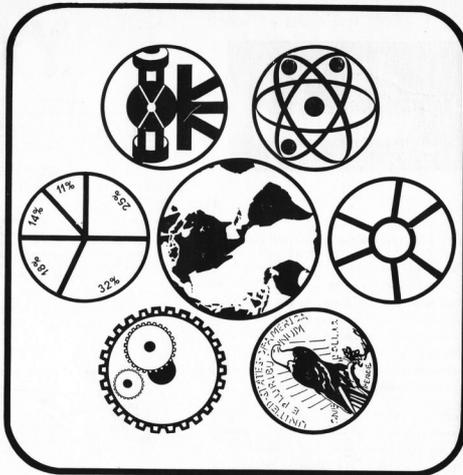
Exempel: Tiden 14,0010 tas som startvärde. Efter decimalomvandling (D,MS) lagras detta värde i minne 01. Konstanten 3600 (utprovad) lagras i minne 00.

Klockan startas sedan genom att man trycker RST/R/S.

Q11d

PROGRAMBITEN 79-3/4

# Programmable T158/59 Specialty Pakettes Electronic Engineering



11040

11040A

## ELECTRONIC ENGINEERING

- **CLASS "X" AMPLIFIER DESIGNED TO SPECIFIED TOLERANCES**  
Calculates the input admittance, transducer gain, and Linvill's C-factor for transistor amplifiers. Current is constrained to a maximum allowable percent collector current variation specified by the user.  
T1-58 only.
- **RF AMPLIFIER ANALYSIS**  
Calculates the input admittance, transducer gain, and Linvill's C-factor for RF amplifiers. Designed for a low frequency analysis.  
T1-58 or T1-59.
- **BIPOLAR JUNCTION TRANSISTOR ANALYSIS**  
Calculates the emitter bypass and coupling capacitors for a B.J.T. circuit.  
T1-58 or T1-59.
- **POWER SUPPLY FILTER DESIGN**  
Load resistance, filter capacitor value, and the minimum, maximum, and average output voltages for both the full- and half-wave rectifier circuits calculated.  
T1-58 only.
- **ZENER POWER SUPPLY DESIGN**  
Calculates the zener and power ratings pertaining to a Zener diode regulated power supply.  
T1-58 only.
- **RESISTIVE VOLTAGE DIVIDER**  
Provides a method to empirically design and evaluate resistive voltage divider networks often encountered in basic designs.  
T1-58 or T1-59.
- **FIELD EFFECT TRANSISTOR ANALYSIS**  
Calculates the gate and drain biasing capacitors for a F.E.T. circuit using the pole zero calculation method.  
T1-58 or T1-59.
- **SOLUTION OF RESISTIVE NETWORKS**  
Solves for the total resistance of both simple and complicated D.C. networks.  
T1-58 or T1-59.

PROGRAMMED MAGNETIC CARDS ARE NOT INCLUDED.  
(The Program Card Data must be stored into blank magnetic cards.)

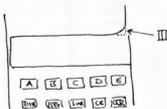
TEXAS INSTRUMENTS

INCORPORATED

DALLAS, TEXAS

TEXAS INSTRUMENTS  
INCORPORATED

Vi har tittat igenom ovanliggande paket, och finner den användbar med några få modifieringar.

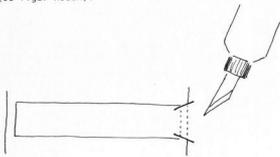


**PRAKTISKA TIPS SOM GÖR PROGRAMMERINGSARBETET SNABBARE...**

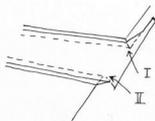
För allt arbete gäller (tyvärr) "Shannons formel", som säger att varje förflyttning tar tid i förhållande till förflyttningssträckans längd, samt i förhållande till den nödvändiga precisionen i rörelsen nära målet. Utgående från denna teori kan ett nytt betraktelsesätt tillämpas på allt arbete som är analytisk synvinkel hittar "tidsödande moment" i en arbetsanalys, där verkligheten är medtagen i beräkningen.

Analysen utfördes med ett TI-59 program på arbetsmoment i samband med användandet av TI-59 och pekade på att det mest tidsödande momentet (i förhållande till nyttan) var att sätta programkortet på plats i slitsen ovanför knapparna. Att sätta in ett kort tar inte bara tid utan kräver dessutom precision och koncentration!

Om räknarens garanti gått ut vågar man kanske göra denna manöver rationellare genom följande "snickeri" (se figur nedan):



1. Först kapas högra "staget" vid korthållaren. Lämpriken används en vass hobbykniv (plasten i räknaren är "snickarvänlig").



2. Därefter fاسas kanterna I, II och III så att kortet lätt glider ned i sitt läge rakt framifrån. Det vill säga, det tidsödande momentet att sticka in kortet i springan undviks.

3. Finslipa kanterna så att kortet lätt går att sätta in som sista figuren visar.



Pl-1

PS



GOTT NUTT RR  
GOTT NUTT RR

# HEXY.

**MASKINSPRÅKS-PROGRAMMERING AV DATORER med hjälp av TI-59**

Av Christer Andreasson

Vid programmering av datorer (hemdatorer) i maskinspråkliga "poker" man byter för byte med en hexadecimal kod.

Det som är tråkigast i denna typ av programmering är att räkna ut adresser enligt vad jag erfart. En adress består av två (2) bytes (dvs den upptar 2 bytes i minnet) och adressering är ofta relativt till programtegensrökarna (PC) med både hopp framåt, bakåt och ---- negativt bakåt (#framst)----

Därför gjorde jag "HEXY" på 59:an för att få adresserna konverterade från decimalt till hexadecimalt (även vice versa).

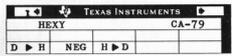
NÅ, ett exempel.

Adress 49152 omvandlas till C000 (hexadec) men ska lagras omvänt, dvs 0000.  
Detta görs enkelt på 59:an: 49152 A.  
Printern skriver ut....  
ADR 0000  
032 11 11 1223 04 4  
033 00 0 124 00 0  
034 42 STD 122 13 13  
035 10 10 126 06 6  
036 01 1 127 42 STD  
037 42 STD 128 19 19  
038 05 05 130 42 STD  
039 01 1 130 42 STD  
040 00 0 131 06 06  
041 00 0 132 01 1  
042 49 PRD 133 00 0  
043 10 10 134 00 0  
044 73 RC+ 135 49 PRD  
045 05 05 136 49 PRD  
046 71 SER 137 43 RCL  
047 00 00 138 15 15  
048 66 66 139 59 INT  
049 44 SUM 140 22 INV  
050 10 10 141 44 SUM  
051 69 DP 142 15 15  
052 25 25 143 65 \*  
053 97 BSZ 143 RCL  
054 09 09 145 19 19  
055 00 00 146 95 =  
056 04 04 147 44 SUM  
057 43 RCL 148 13 13  
058 10 10 149 01 1  
059 69 DP 150 06 6  
060 04 04 151 42 INV  
061 43 RCL 152 49 PRD  
062 20 20 153 19 19  
063 69 DP 154 97 BSZ  
064 06 06 155 06 06  
065 16 A\* 156 01 01  
066 85 + 157 32 32  
067 01 1 158 43 RCL  
068 = 159 13 13  
069 32 X:T 160 61 GTO  
070 08 8 161 11 A  
071 42 INV 162 76 LBL  
072 77 GE 163 16 A\*  
073 00 00 164 93 \*  
074 77 77 165 00 0  
075 32 X:T 166 00 0  
076 92 RTN 167 00 0  
077 01 1 168 01 1  
078 05 5 169 49 PRD  
079 22 INV 170 10 10  
080 77 GE 171 43 RCL  
081 00 00 172 10 10  
082 73 RC+ 173 22 INV  
083 32 X:T 174 59 INT  
084 85 + 175 22 INV  
085 02 2 176 44 SUM  
086 95 = 177 10 10  
087 92 RTN 178 65 \*  
088 32 X:T 179 01 1  
089 85 + 180 52 EE  
090 05 5

Ska man hoppa tex 25 steg bakåt får man hexkoden för 25 genom samma procedur, men adressutskriften är i detta fall ointressant. Ska man hoppa 25 steg negativt bakåt fås koden för -25 genom att trycka i 25 B. Även här är adressutskriften ointressant. (25= 19 (hex), -25= E7 (hex) )  
Observera att minustecknet inte matas in.

DEC TILL HEX	
Inmatningskod	
00 =	00
01 =	01
02 =	02
03 =	03
04 =	04
05 =	05
06 =	06
07 =	07
08 =	08
09 =	09
10 =	A
11 =	B
12 =	C
13 =	D
14 =	E
15 =	F

Exempel:	
Konvertering av FF 03 (hex) (adr.=03 FF) till decimalt:	
F F 0 3	
15150003 C	
Svar: 65283 (dec)	
DEC	HEX A
NEG DEC	HEX B
HEX	DEC C
Initialisering	E
Obs.	
PC 100 nödvändig.	



000	76	LBL	091	95 =	181	00	0
001	11	R	092	92 RTN	182	08	8
002	42	STD	093	76 LBL	183	22	INV
003	20	20	094	12 B	184	52	EE
004	42	STD	095	45 STD	185	05 =	
005	05	05	096	05 25	186	44	SUM
006	04	4	097	02 2	187	10	10
007	42	STD	098	05 5	188	01	1
008	06	06	099	06 6	189	03	3
009	42	STD	100	75 =	190	01	1
010	09	09	101	43 RCL	191	06	6
011	01	1	102	25 25	192	03	3
012	06	6	103	05 =	193	05 =	5
013	22	INV	104	61 GTO	194	06	6
014	49	PRD	105	11 A	195	02	2
015	05	05	106	76 LBL	196	69	DP
016	43	RCL	107	43 C	197	03	3
017	05	05	108	42 STD	198	43	RCL
018	22	INV	109	14 14	199	10	10
019	05	05	110	42 STD	200	05	5
020	22	INV	111	08 8	201	04	4
021	44	SUM	112	22 INV	202	69	DP
022	05	05	113	28 LDG	203	05	05
023	65	*	114	EE	204	91	R/S
024	01	1	115	22 INV	205	76	LBL
025	06	6	116	52 EE	206	15	E
026	95 =	=	117	95 =	207	25	CLR
027	06	6	118	42 STD	208	47	CM2
028	06	06	119	15 15	209	29	CP
029	97	BSZ	120	25 CLR	210	69	DP
030	06	06	121	02 STD	211	00	00
031	00	00	122	13 13	212	02	2
032	11	11	123	04 4	213	03	3
033	00	0	124	00 0	214	01	1
034	42	STD	125	09 9	215	07	7
035	10	10	126	06 6	216	04	4
036	01	1	127	42 STD	217	04	4
037	42	STD	128	19 19	218	04	4
038	05	05	129	09 9	219	05	5
039	01	1	130	42 STD	220	69	DP
040	00	0	131	06 06	221	02	02
041	00	0	132	01 1	222	01	1
042	49	PRD	133	00 0	223	05	5
043	10	10	134	00 0	224	01	1
044	73	RC+	135	49 PRD	225	03	3
045	05	05	136	49 PRD	226	02	2
046	71	SER	137	43 RCL	227	00	0
047	00	00	138	15 15	228	10	10
048	66	66	139	59 INT	229	08	8
049	44	SUM	140	22 INV	230	04	4
050	10	10	141	44 SUM	231	02	2
051	69	DP	142	15 15	232	69	DP
052	25	25	143	65 *	233	04	04
053	97	BSZ	143	43 RCL	234	69	DP
054	09	09	145	19 19	235	05	05
055	00	00	146	95 =	236	98	ADV
056	04	04	147	44 SUM	237	25	CLR
057	43	RCL	148	13 13	238	91	R/S
058	10	10	149	01 1			
059	69	DP	150	06 6			
060	04	04	151	42 INV			
061	43	RCL	152	49 PRD			
062	20	20	153	19 19			
063	69	DP	154	97 BSZ			
064	06	06	155	06 06			
065	16	A*	156	01 01			
066	85	+	157	32 32			
067	01	1	158	43 RCL			
068	=	=	159	13 13			
069	32	X:T	160	61 GTO			
070	08	8	161	11 A			
071	42	INV	162	76 LBL			
072	77	GE	163	16 A*			
073	00	00	164	93 *			
074	77	77	165	00 0			
075	32	X:T	166	00 0			
076	92	RTN	167	00 0			
077	01	1	168	01 1			
078	05	5	169	49 PRD			
079	22	INV	170	10 10			
080	77	GE	171	43 RCL			
081	00	00	172	10 10			
082	73	RC+	173	22 INV			
083	32	X:T	174	59 INT			
084	85	+	175	22 INV			
085	02	2	176	44 SUM			
086	95	=	177	10 10			
087	92	RTN	178	65 *			
088	32	X:T	179	01 1			
089	85	+	180	52 EE			
090	05	5					

EXEMPEL  
PA UTSKRIFT

HEXY CA-79  
65535. FFFF  
HEXY ADR: FFFF  
65283. FF03  
HEXY ADR: 03FF  
131. 0083  
HEXY ADR: 8300

Pl-11



# NÅGOT OM TI-59'S DATOR-LIKNANDE EGENSKAPER

Eftersom TI-59 har "gemensamt" minne för program och data ligger det nära till hands att undersöka om detta förhållande kan användas vid programmering av maskinen.

Uppsatsen här är ganska kortfattad, och skall enbart ses som ett underlag för fortsatta experiment och undersökningar för den som är intresserad av detta.

Om vi lagrar talet 9.208821476 i register 59 och trycker på 5 2nd Op 17, finner vi att följande lilla programsnitt är lagrad i minnet:

```
483 76 2nd Lbl
484 14 D
485 82 HIR
486 08 08
487 92 INV SBR
```

Sist i uppsatsen finns ett program som kan användas för inmatning av instruktionskoder. Jag visar hur programmet skall användas på ovanstående programsnitt.

Mata in	Tryck på	Sifferindikator	Anmärkning
1	SBR	2nd Pgm 99	Initiering
2	76	E 7.6	1:a instr.
3	14	E 1.476	2:a instr.
4	82	E 8.21476	3:e instr.
5	08	E 0.8821476	4:e instr.
6	92	E 9.208821476	5:e instr.
7	2nd E	98	Nytt reg.

Rutinen 2nd E lagrar talet som finns i x-registret i register 99 och ställer om registerpekaren på register 98. Nu kan fler instruktionskoder matas in.

Detta visar att det är fullt möjligt att använda denna teknik för programmering av maskinen. Programsnittet ovan är kort och rymms i endast ett register. För längre program måste flera register användas. Det är här svårigheterna börjar. Programmet måste delas upp i segment om maximalt 6 programsteg, fler rymms inte i ett register. Denna uppdelning kan inte göras hur som helst. Förutom TI-59's egna språkregler måste man känna till följande två viktiga regler:

**Regel nr 1:** Instruktionssekvensen får inte resultera i ett tal som är mindre än 1, eller större än 10.

Antag att vi "packat" ett antal programsteg i ett register. Denna "packe" motsvaras alltså av ett tal i registret. Detta tals 10-exponent måste alltid vara lika med noll, dvs talets storlek måste alltid ligga mellan 1 och 10. Använder man inmatningsprogram som presenterades ovan blir talets storleksordning alltid mindre än 10. Ibland, t.ex. om sista programsteget i en "packe" innehåller en adress, kan talet bli mindre än 1. Detta kan avvjälpas om man lägger en 2nd Nop-instruktion (68) sist i "packen".

**Regel nr 2:** Skarven mellan två register medför att talet i x-registret försvinner vid programkörning.

I varje register finns det ett visst minnesutrymme som upptas av 10-exponent och teckenbitar. Detta minnesutrymme är inte åtkomligt för inmatning av program enligt den här metoden. Dessa delar i minnet innehåller idel nollor, och motsvaras också av nollor i programminnet. Vid programkörning matas dessa nollor upp i x-registret, varvid det förutvarande talet i x-registret försvinner. Om detta tal skall användas för vidare beräkningar, måste "packens" sista instruktion vara en undanlagringsinstruktion. För undanlagring är t-registret det lämpligaste.

Båda dessa regler gör att programminnet utnyttjas väldigt dåligt vid programmering med den här metoden. Ibland kan ett register enbart innehålla två effektiva programsteg.

På sidan IV-74 i instruktionsboken finns ett program för beräkning av N! faktulet. Utan att skriva om programmet gick det bra att mata in instruktionerna direkt i maskinen med hjälp av inmatningsprogrammet och de två ovanstående reglerna. Programmet fungerade genast felritt, men eftersom det tog ca 80 programsteg i anspråk var det ganska långsamt.

Vad är då nyttan med den här tekniken? Det är fullt klart att den inte är särskilt lämplig att mata in färdiga program med. Följande program exempel beskriver ett sätt:

Subrutinen C ger användaren tillgång till HIR operationerna, men med indirekt adressering. Operanden lagras i t-registret och adressen i x-registret.

```
000 53 C          021 09 9
001 24 E         022 02 2
002 55 +         023 95 =
003 01 1        024 55 =
004 00 0        025 01 1
005 00 0        026 00 0
006 34 >        027 95 =
007 92 RTH      028 42 STD
008 76 LBL      029 59 59
009 13 C        030 05 5
010 85 +        031 69 DP
011 08 8        032 17 17
012 02 2        033 32 MIT
013 71 SBR      034 71 SBR
014 00 00       035 04 04
015 00 00       036 85 85
016 95 =        037 32 MIT
017 71 SBR      038 06 6
018 00 00       039 69 DP
019 00 00       040 17 17
020 85 +        041 32 MIT
                042 92 RTH
```

Exempel: Beräkna  $3 + 10 - 2 \times 14/7 =$

Mata in Tryck ned Mata in Tryck ned Sifferindikator

3	x=+	8	C	3
10	x=+	38	C	10
2	x=+	7	C	2
14	x=+	47	C	14
7	x=+	67	C	7
		17	C	4
	x=+	58	C	4
		18	C	9

Korrekt svar = 9.

Detta är alltså ett exempel på hur ett program kan fås att skapa nya program, som sedan kan användas som subrutiner. Man kan även tänka sig rutiner som går in i program och ändrar enkasta programsteg, t.ex. byter operationer (+ mot -, eller liknande), tar bort instruktioner eller skriver in nya (t.ex. Labels, som motsvarar de användardefinierade tangenterna, kan då flyttas omkring godtyckligt i programminnet.)

Man bör även skriva s.k. överstättande program. Det är fullt tänkbart att ett program (i en programmodul, vilket också finns) kan översätta programkoder för HP-67 till programkoder för TI-59, och som dessutom även tar hänsyn till räknedornas olika maskinspråks uppbyggnad. Här nämras vi oss, vad jag tror är nästa generations räknedor - de som blir programmerbara i högnivåspråk.

Program för inmatning av operationskoder.

000 55 +	023 10 E*
001 01 1	024 72 ST*
002 00 0	025 01 01
003 95 =	026 69 DP
004 92 RTH	027 31 31
005 76 LBL	028 00 0
006 15 E	029 42 STD
007 32 MIT	030 00 00
008 43 RCL	031 43 RCL
009 00 00	032 01 01
010 71 SBR	033 92 RTH
011 00 00	034 76 LBL
012 00 00	035 36 PGM
013 85 +	036 01 1
014 32 MIT	037 00 0
015 95 =	038 69 DP
016 71 SBR	039 17 17
017 00 00	040 47 CMS
018 00 00	041 09 9
019 42 STD	042 09 9
020 00 00	043 42 STD
021 92 RTH	044 01 01
022 76 LBL	045 92 RTH

Hur inmatningen tillgås, ser man exempel i början av uppsatsen. När inmatningen är klar, har man möjlighet att trycka på 2nd CP, varvid programminnet raderas och de labels som var upptagna av inmatningsprogrammet frigörs för användning i det inmatade programmet. Först därefter ställer man om minnesuppdelningen som man önskar.

PIPI



# TEKNIKER KONSTRUKTÖR



**Texas Instruments TI-58/59 erbjuder dig 80 olika metoder att underlätta ditt arbete**

TI-58/59 arbetar med Texas Instruments unika modulteknik (Solid State Software). Inom ämnesområdena elteknik, hälftahteknik, matematik och statistik finns 5 olika moduler. Varje modul innehåller 5 000 programtext.

TI-58/59 med tillbehör är inte bara ett av världens mest avancerade fickdatorsystem. Kombinationen fickräknare — programmoduler är också mycket lätt att använda — även utan tidigare erfarenhet. Vill du ha utskrift av siffror och bokstäver, anslut skrivaren PC-100. Då kan du också plotta kurvor och histogram, förtydliga programmen med symboler och ledtexter, lista dataregister och program m m.

Du kan också komplettera modulprogrammen med egna program. Med en TI-58, kan du dessutom lagra egna program och data på magnetkort.

Texas Instruments har idag totalt ca 800 program inom olika ämnesområden, så av dessa behandlar t ex: Linjär elasticitetslära, matris-

Extramoduler till TI-58/59C/59  
Flygnavigation, Marinavigation, Statistik, Finans, Matematik, Elteknik, Lantmäteri (amerikansk), Fastighetsinvestering, Värdepappersinvestering, RPN-Simulator, Spel, Lantbruk, Byggnadsteknik, Lantmäteri (Europeisk)

#### Programböcker

(Innehåller vardera 6-11 listade program)  
Astrologi, Byggnadsteknik, Infrastrålning, Spel, Ols/Gas/Energiberäkningar, Statistik, Matematik, Säljekonst, Produktionsplanering, Lab/Kemi, Utnyttjande av skrivaren, Datorkonverteringar, Vätskeberäkningar, Elteknik/Elektronik, 3D figurer, Värdepappersinvestering

beräkningar, geometri, integrering, differentialekologi, komplexa funktioner, ej-filtterberäkningar, omvandlingar, statistiska funktionsfunktioner, test och stickprov, linjär och icke linjär regressionsanalys, histogramkonstruktion. Låt oss titta på hur ett av dessa program kan underlätta ditt arbete:

Du skall beräkna en balk med 8 stödpunkter. Balken har varierande böjstyvhets och utsätts för olika typer av belastningar. Tvärkrafter och moment vid samtliga stödpunkter sökes. Ur modulen "Structural Engineering" använder du programmet "Continuous Beam". Mata in nödvändiga indata — på skrivaren får du kvittens och omedelbart de sökta svaren — med ledtexter och symboler.



## TEXAS INSTRUMENTS

Fack 100 54 Stockholm Tel. 08-23 54 80

Ca pris: 250,-

Ca pris: 360,-

Ca pris: 580,-

Ca pris: 59,-



#### SMÅDATORMARKNADEN INFÖR 80-talet

Marknaden för smådatorsystem, SD (persondatorer, mikrodatorsystem, hemdatorer etc), är ny. Den kom igång i slutet av 1975 i USA och 1978 i Sverige. På 2 år har ca 8 000 SD sålts i Sverige. I USA är siffran uppe på 6-700 000 SD. Prognoser pekar på fortsatt snabb ökning.

En typisk SD består i grundutförandet av bildskärm, tangentbord inklusive själva datorn och en kassettbandspelare. Den programmeras vanligen i det enkla och lättlärdade undervisningspråket BASIC och kostar under 10 000 kr. De vanligaste förekommande kringutrustningarna är skrivare och sk k floppy disk (flexdiskminne). En SD med skrivare och floppy disk som utgör standardutrustning för de flesta professionella tillämpningar, kostar normalt ej över 25 000 kr.

Marknaden i Sverige för SD är i huvudsak uppdelad på följande märken:

ABC 80 65 - 70 %

Commodore PET ca 20 %

APPLE ca 5 %

och övriga

(TRS 80, Cromemco, Storcercer, Scientific Ohio etc)

Det bör påpekas att detta är något idag. Flera mycket stora företags förväntas etablera sig på

marknaden inom några år. Texas Instruments kommer troligen att börja marknadsföra sin smådator vid mitten av 1980, vilket förväntas få stor betydelse för utvecklingen framöver.

Hittills har hårdvarumarknaden dominerat. Av de ca 8 000 SD som sålts i Sverige har endast ett mindre antal levererats med mjukvara (d v s program). Programvarumarknaden ökar för närvarande mycket starkt.

(Jfr SIND, Datormarknaden inför 80-talet.)

Programutvecklingen sker nu på bred front i landet av konsulter och användare främst inom områdena utbildning, teknik/vetenskap och inte minst administrativa tillämpningar. Detta ökar utbildningsbehovet både för programmering och användning av SD. Behovet är särskilt stort när det gäller elementära BASIC-kurser och applikationsinriktade kurser, t ex användning av SD med program för bokföring, ordbehandling, registerhantering etc.

Traditionellt har datorleverantörerna svarat för en stor del av utbildning och installation av datorsystem hos kunden. Situationen för SD är helt annorlunda. Det låga priset på maskinvaran gör att priset även på programvaran måste hållas lågt. Detta synar uppmotkar av standardprogram. Leverantörerna av SD har normalt varken resurser eller tillräcklig kunskap att utbilda det snabbt växande antalet användare av standardprogram. SD säljs vanligen utan installation hos kunden, ungefär som kontorsmaskiner, med 1 års garanti.

## UTBILDNING PÅ SMÅDATORER



#### 80-talets ABC-kurser

Datarevolutionen står för dörren. Under 80-talet kommer smådatorer att bli ett lika naturligt arbetsredskap som skrivmaskinen är idag. Denna framtida utveckling och det faktum att många redan idag använder smådatorer i arbetet gör att behovet av utbildning snabbt ökar.

Dårför presenterar vi nu marknadens bredaste sortiment av utbildning på smådatorer. Ett heltäckande program med ett 30-tal kurser av varierande innehåll och svårighetsgrad från rena nybörjarkurser till specialistkurser.

Sänd efter vår kurskatalog!

Innehåll:

- Företagsinterna kurser
- Allmänbildningskurs ABC om smådatorer
- Programmeringskurser i BASIC
- Grundkurs i BASIC
- Fortsättningskurs i BASIC
- Avancerad kurs i BASIC

Användarkurser

- Bokföring
- Adresssystem
- Ordbehandling
- Statistik
- Övriga kurser
- Grundkurs i datorteknik
- Terminalanvändning
- Assemblerprogrammering

För ytterligare information ring 08-753 31 40.

Insändes till LEXICON AB, Box 136, 182 12 DANDERYD

Sänd kurskatalogen för VT 1980

Namn

Adress

Postadress



Här kommer ytterligare en metod för uppmärkning av magnetkort som komplettering till den tidigare artikeln i ämnet som publicerades i PROGRAM BITEN 79-2.

Metoden är screentryck, ett urgamalt förfarande som tillämpas vid tillverkning av skyltar, decalor, tryck på tyger o s v. Själva tryckningen kan utföras hemma i köket - om man får för hustrun förstås...

En del grejor behövs:

- 1 st screenduk med ram
- 1 st sats emulsion för beläggning av duken
- 1 st rakel
- 1 st rulle mask tape
- färg
- rengöringsmedel



Det här medför en viss investering men i gengäld kan man använda utrustningen många gånger.

FÖRST ETT ORIGINAL...

Man utgår från ett original som antingen kan gnuggas eller maskinskrivas, helst på elektrisk maskin försedd med karbonband. Trycket måste vara jämnt och fint. En lämplig stilsort är IBM:s ORATOR - där man använder utslutande de små bokstäverna. Sifferorna på ORATOR är för stora och måste man ha sådana med får man skriva just dessa med någon annan kula.

Sedan fotograferas det färdiga originalet i exakt rätt storlek för att passa magnetkortet. Filmen skall vara litofil som är absolut svart-vit utan mellantoner.

Denna film skall sedan "vändas" en gång på ny film så att texten blir svart precis som den skall vara på det färdiga kortet. Just den här biten i processen kan vara besvärlig om man saknar lämplig utrustning men kanske fotografen på orten kan hjälpa till.

# SCREEN TRYCK

- metod för perfektionister...

SCHABLONEN GÖRES I ORDNING...

Först skall ramens tyg beläggas med ljuskänlig beläggning och detta måste göras i mörkrum med rött eller grönt ljus. Har man eget fotolabb hemma är det lämpligt att hålla till där. När beläggningen är klar och har torkat lägges ramen med undersidan upp på en bit jämn skumgummiplatta och filmen placeras spegelvänd med en glas-skiva över och belyses med en kvicksilverlampa.

Denna skall vara punktformig och belysningstiden bör provas ut. Ett närmvärde är ca: 90 cm från lampen och en belysningstid på ungefär fem minuter. När detta är gjort skall ohärdad beläggning spolas bort med 40-gradigt vatten. De ytor som varit täckta av filmens text blir öppna och kan sedan släppa igenom färg. Ramen lägges plant och får torka. Utanför den täta delen av schablonen tåtas sedan hela ramen med tape så att inte färg kan rinna igenom.

ORDNINGSTÄLLANDE FÖR TRYCKNING...

Ramen sättes fast ledbart på två helt glappfria gängjärn med duken nedåt och så att den lätt kan fällas upp. (Fönsterhakar av nylon med led är 100% glappfria och utmärkt lämpliga). Kortet som skall tryckas passas in under ramen och fästes med dubbelhäftande tape vid underlaget som måste vara absolut plant. När så ramen fälls ned skall texten på schablonen passa exakt i de förtryckta fälten på kortet. Dukens undersida skall i nödfall ligge hanna två mm ovanför kortet och ramen får pallas upp till lämplig höjd med kartongbitar. När man sedan drar rakel över tyget kommer detta att fjädra ner precis i kontakt med kortet.

KLART FÖR TRYCKNING...

Nu är det bara att hålla litet färg i ramen och dra rakel EN gång över duken. Håll rakel i ca: 45 graders lutning och tryck inte för hårt. Duken skall tryckas ned precis så mycket att den "doppar" ned i sin undersida till ytan som skall tryckas. Har man gjort rätt blir resultatet ett underbart fint tryck. Förigen känns heltort redan efter en kvart men måste torka flera dagar innan kortet kan sättas in i sin plastficka.



Efter tryckning av valfritt antal kort bryr ramen genast rengöras från färg och schablonen avlägsnas.

AB Hagberg & Winholm, S:t Eriksgatan 101, 113 31 Sthlm tel 08-7360190, har alla nödvändiga attiraljer som kan behövas och där står man också gärna till tjänst med råd. Div detaljerade anvisningar medföljer också t.ex. förpackningen med den ljuskänliga emulsionen.

Screentrycksförfarandet kan verka besvärligt och omständigt men slutresultatet blir outstanding! Vi vill värt allt arbete om man är perfektionist. Och vilken matematiker är inte det?

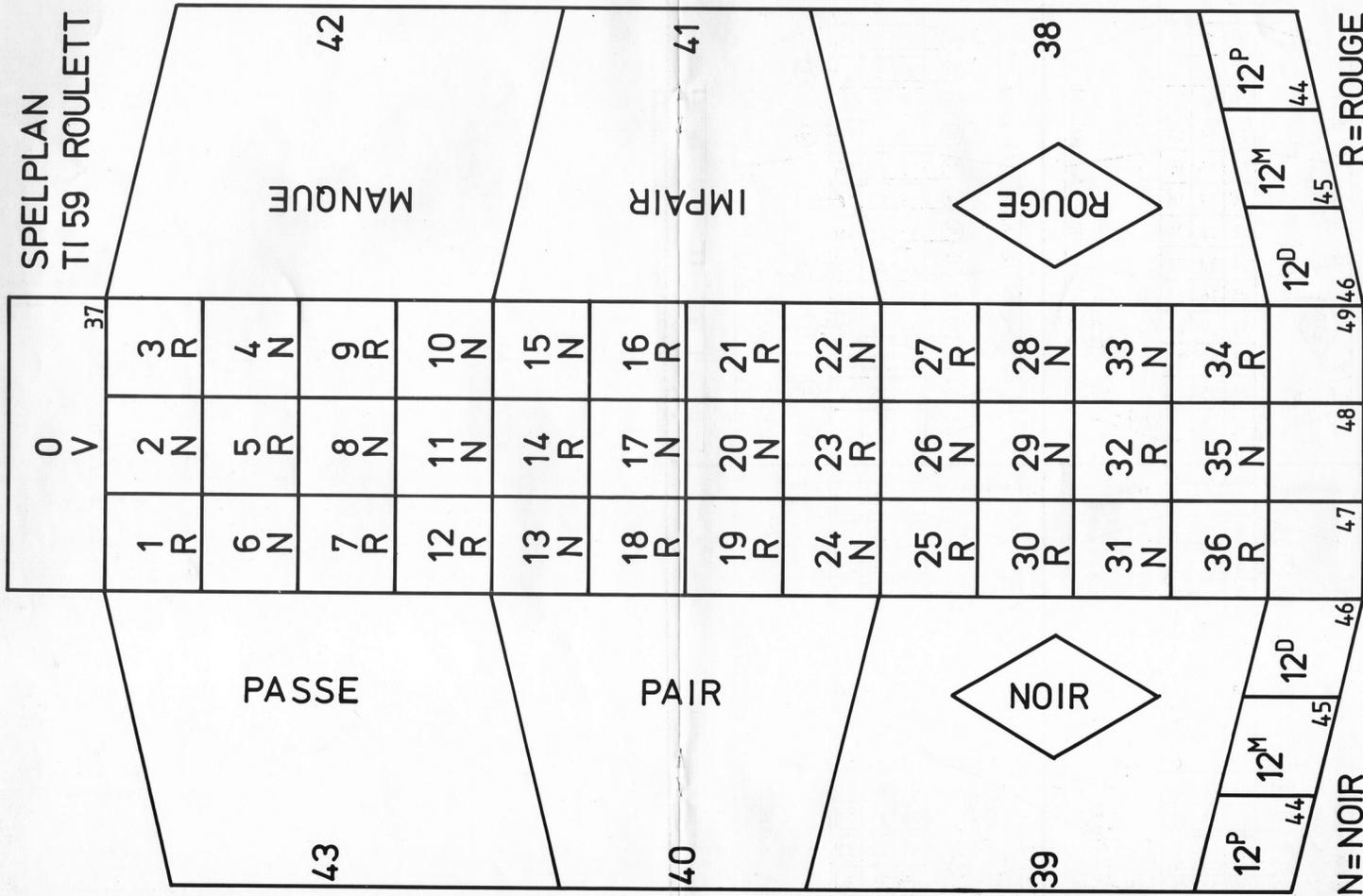
UX

PROGRAMBITEN 79-3/4

PROGRAMBITEN 79-3/4

ATT RIVA UT SPELPLANEN PASSAR PROGRAMMET, MEN SKILTER SIG FRÅN 'STANDARD'.

SPELPLAN  
TI 59 ROULETT



			0	37				
PASSE	1	2	3	42				
	6	5	4					
	7	8	9					
	12	11	10					
PAIR	13	14	15	41				
	18	17	16					
	19	20	21					
	24	23	22					
NOIR	25	26	27	38				
	30	29	28					
	31	32	33					
	36	35	34					
12 <sup>P</sup>	12 <sup>M</sup>	12 <sup>D</sup>				12 <sup>D</sup>	12 <sup>M</sup>	12 <sup>P</sup>

			0	37	RYTTARE (HÄST) TVÅ NUMMER			
PASSE	68	1	2	3	42	69		
	70	6	5	4				
	71	7	8	9				
	73	12	11	10				
PAIR	74	13	14	15	41	75		
	76	18	17	16				
	77	19	20	21				
	79	24	23	22				
NOIR	80	25	26	27	38	81		
	82	30	29	28				
	83	31	32	33				
	85	36	35	34				
12 <sup>P</sup>	12 <sup>M</sup>	12 <sup>D</sup>				12 <sup>D</sup>	12 <sup>M</sup>	12 <sup>P</sup>

			0	37	TVÄRS TRE NUMMER EN RAD			
PASSE	56	1	2	3	42			
	57	6	5	4				
	58	7	8	9				
	59	12	11	10				
PAIR	60	13	14	15	41			
	61	18	17	16				
	62	19	20	21				
	63	24	23	22				
NOIR	64	25	26	27	38			
	65	30	29	28				
	66	31	32	33				
	67	36	35	34				
12 <sup>P</sup>	12 <sup>M</sup>	12 <sup>D</sup>				12 <sup>D</sup>	12 <sup>M</sup>	12 <sup>P</sup>

			0	37	TVÄRS SEX NUMMER TVÅ RADER			
PASSE	50	1	2	3	42			
	51	6	5	4				
	52	7	8	9				
	53	12	11	10				
PAIR	54	13	14	15	41			
	55	18	17	16				
	56	19	20	21				
	57	24	23	22				
NOIR	58	25	26	27	38			
	59	30	29	28				
	60	31	32	33				
	61	36	35	34				
12 <sup>P</sup>	12 <sup>M</sup>	12 <sup>D</sup>				12 <sup>D</sup>	12 <sup>M</sup>	12 <sup>P</sup>

## Litteratur om programmerbara miniräknare

Vi fortsätter föregående bokrecension med att nämna ett antal böcker som föreningen fått kännedom om.

För den som befinner sig på nybörjarnivå finns böcker som behandlar programmering "ifrån grunden" som UNDERSTANDING CALCULATOR MATH utgiven av TI Learning Center, PROGRAMMABLE CALCULATORS - BUSINESS APPLICATIONS av Aronofsky-Fraser-Greyndolls (TI-57, SR 52, 56, 60 samt MBA). Pris ca 79 kr inkl moms.

CALCULATOR ANALYSIS FOR BUSINESS AND FINANCE utgiven av Mc Graw-Hill som är skriven för TI:s MBA (programmerbar affärsräknare) innehåller många kvalificerade program. För en 58/59-ägare med stort intresse för affärsprogram kan den klara uppbyggnaden kring olika affärsberäkningar (ca 200 ex!) vara till hjälp vid konstruktion av program där mastermodulens affärs/finansprogram utnyttjas. Pris ca 45 kr inkl moms.

En annan amerikansk bok om grunderna i programmering är PROGRAMMABLE CALCULATORS: HOW TO USE THEM av Sippl & Sippl (McGraw-Hill), \$ 16 i USA).

HOW TO PROGRAM YOUR PROGRAMMABLE CALCULATOR av Snover & Spikell (Spectrum) är ny för i år (\$ 8 i USA). PROGRAMMING PROGRAMMABLE CALCULATORS av Engelson (Hayden Books Co) påstår sig använda en lättfattligare jargong än handböckerna när program för SR 52, 56, TI-57/58/59, PR 100 och APF's programmerbara räknare presenteras.

Vidare finns det en mera "hårdvaruorienterad" bok - MINI-PROCESSORS: FROM CALCULATORS TO COMPUTERS av Heiserman (McGraw-Hill). Den beskriver hur ett system byggs upp med enkla räknare (e) programmerbara) till ett slags programtyrd räknare. THE "COMPUTATOR" BOOK av samme författare påstår sig ge tips om "hundratals sätt att använda räknarkretsar" med instruktion om hur elektroniken byggs upp.

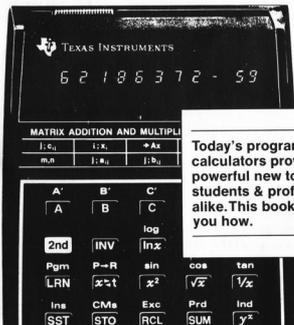
Den bok som främst rekommenderas för föreningens medlemmar är SOURCEBOOK FOR PROGRAMMABLE CALCULATORS av TI Learning Center (TI-USA) som bedömts vara av sådant intresse att vi visar innehållsförteckningen som avslutning på denna spalt. Pris 79 kr inkl moms.

Programmen i boken presenteras klart och koncist. De program som imponerade mest på mig var "Bode/Nykvist-plot", där både db-tal, fasvinkel,  $H_p$ ,  $H_c$  och  $H$  anges för efterfrågade frekvenser.



# Sourcebook for Programmable Calculators

Step-by-step programmed solutions, whatever your field:  
 ■ mathematics ■ calculus ■ statistics ■ business and operations research ■ economics ■ biology ■ engineering ■ physics and astronomy ■ music ■ and much more... designed for use with the TI Programmable 58 and 59 calculators.



Today's programmable calculators provide a powerful new tool for students & professionals alike. This book shows you how.

## Table of Contents

<b>INTRODUCTION</b>	
Calculator Evolution	1-1
Calculator Roles in Today's College Studies	1-3
Programmability	1-4
Some Specifics	1-4
How to Use This Book	1-6
Notes About Your Calculator	1-7
<b>1. BASIC NUMBER THEORY</b>	
Introduction	1-1
Sequences and Series	1-2
Greatest Common Divisor	1-6
Prime Number Testing	1-9
Prime Factorization	1-12
Base Conversion	1-16
<b>2. COLLEGE ALGEBRA AND TRIGONOMETRY</b>	
Introduction	2-1
Graphing Functions	2-2
Polynomial Evaluations	2-7
Systems of Linear Equations (Master Library Program)	2-10
Quadratic Equations	2-15
Trig and Triangles	2-19
Trig Functions	2-22
<b>3. CALCULUS AND LINEAR ALGEBRA</b>	
Introduction	3-1
Approximating Roots of a Function	3-2
Numerical Approximation of Derivatives	3-6
Definite Integrals—Trapezoidal Rule	3-10
Runge-Kutta Method (Systems of Differential Equations)	3-14
Curve-Fitting with Least-Squares Approximation	3-20
Systems of Linear Equations (Row Reduction)	3-33
<b>4. STATISTICS</b>	
Introduction	4-1
Descriptive Measures	4-2
The Normal Curve	4-10
The Standard Normal Curve	4-15
t-Distribution	4-19
Linear Regression and the Correlation Coefficient	4-24
Analysis of Variance	4-31
Tables	4-36

<b>5. MUSIC THEORY</b>	
Introduction	5-1
Intervals	5-3
Chord Identification	5-3
Sight-Singing and Melodic Contours	5-13
Composition	5-20
Pitch Frequency	5-26
Harmonics	5-30
Selected Listening Guide for Computer and Electronic Music	5-33
<b>6. BUSINESS AND OPERATIONS RESEARCH</b>	
Introduction	6-1
Inventory Control	6-5
Variable Periodic Cash Flows and Internal Rate of Return	6-5
Capital Budgeting	6-12
Queuing Model	6-20
Markov Analysis	6-25
<b>7. ECONOMICS</b>	
Introduction	7-1
Seasonal Index (Ratio to Centered Average)	7-3
Reference Cycle	7-9
Fluctuation Around the Trend Line	7-15
Interaction of Multiplier and Accelerator	7-23
Short-Run Production Function and Cost Curves	7-28
<b>8. BIOLOGY</b>	
Introduction	8-1
Genetics	8-2
Blood-Type Matching	8-8
Population Growth	8-15
Sedimentation	8-21
<b>9. BIOMEDICAL ENGINEERING</b>	
Introduction	9-1
Active Filter to Improve Signal-to-Noise Ratio	9-7
Estimating Number of Samples for Specified Standard Error	9-9
Transducer Scaling and Interactions	9-10
Cardiac Output Using the Fick Equation	9-19
Student's t-Test	9-21
Regression Analysis	9-30
<b>10. ELECTRICAL AND SYSTEMS ENGINEERING</b>	
Introduction	10-1
Series-Parallel Impedance	10-2
Smith Chart Calculators	10-6
Transistor Parameter Conversion	10-16
Bode/Nyquist Calculations	10-21
Root-Locus Calculations	10-28
Step and Pulse Response	10-35
Convolution	10-39
Discrete-Time System Response	10-44
<b>11. PHYSICS AND ASTRONOMY</b>	
Introduction	11-1
Framhofer Diffraction at a Circular Aperture	11-2
Feyn's Method and Studies of Motion	11-8
Central Force Motion	11-16
Two-Dimensional Speed Distribution	11-25
The Eruption of Time	11-37
Sundial Design	11-43
<b>PS</b>	

Lennart Råde har givit ut följande böcker om programmering:

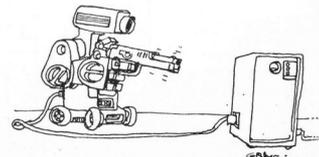
- Att chansa med räknadosan-Del 1. Diskreta problem.
- Att chansa med räknadosan-Del 2. Kontinuerliga problem.
- Att chansa med räknadosan-Del 2. Kontinuerliga problem.
- Att programmera miniräknare.
- Att programmera Texas TI-57.
- Att programmera CASIO 501 och 502.

Ur boken "Att programmera miniräknare" hittade vi ytterligare litteratur:

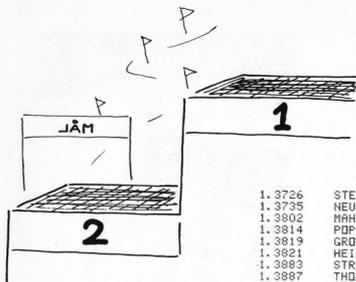
- A. Deledicq, P. Vaschalde, Le calculateur programmable de Poche. (Cedic, Paris 1978).
- R. M. Eisberg, Applied mathematical Physics with Programmable pocket Calculators, (McGraw Hill, NY 1976).
- R. M. Eisberg, W. Hyde, Countdown: Skydivers, Rocket and Satellite motion on Programmable Calculators, (Dilithium Press, Portland, 1979).
- C. J. Sippl, Programmable Calculators (Matrix Publishers, Champaign, 1977).
- S. L. Snover, M. A. Spikell, How to program your programmable calculator, (Prentice Hall, Englewood Cliffs 1979)



Programmerbar affärsräknare (32 steg) - MBA  
 PROGRAMTAGEMÅNEN:  
 LRN  
 RST  
 R/S  
 SST



VAD KAN DETTA SYFTA PÅ?  
 SVAR I NÄSTA NUMMER!



1.3726 STEN  
1.3735 NEUR  
1.3802 NAHR  
1.3814 POPA  
1.3819 GRDS  
1.3821 HEID  
1.3883 STRR  
1.3887 THEE

Detta bidrag har vi fått från Henrik Tjernberg, studerande på KTH. Han använder i det här programmet en speciell märkningsmetod som sammankopplar två värden oberoende av i vilket minne det ena värdet är lagrat. En metod som kan vara användbar även i andra sammanhang.

Programmet sorterar tider (el. liknande) samtidigt som det håller reda på fyra tillhörande alfameriska tecken.

Till exempel kan det vara praktiskt vid idrottstävlingar av olika slag.

### "Brommärkning"

För att rätt text skall kunna sammankopplas med tillhörande tid, "märks" tiden med numret på det minne i vilket texten lagrats. Det sker genom att numret läggs efter det antal decimaler i tiden som man angivit vid initieringen.

Ex. 1.572654  
Tid Minnesnr. 54  
I min. 57 s. 26 100-delar

För att programmet skall fungera får antalet heltalsiffror + decimaler inte överstiga 9. Maximala antalet tider som kan sorteras är 46. (71-59)  
Sorteringen sker genom "intervallhalveringsmetoden".

Den går så till att den tid som skall sorteras in jämförs med den tid som ligger mitt i den redan sorterade listan.

Beroende på utfallet av jämförelsen delas sedan listans övre el. undre halva i mitt- och en ny jämförelse görs. Detta upprepas tills dess man funnit den plats i listan där tiden skall lagras.

Partitionering: 10 OP 17		
Sortering av tider-med tillh. text		
Tid	Lista	Dec. Init

### INITIERING

080 76 LBL  
081 15 E  
082 47 CMS  
083 42 STD Antal decimaler  
084 99 99  
085 95 +  
086 02 2  
087 95 =  
088 94 +/-  
089 42 INV  
090 28 LDG  
091 42 STD Multiplikationsfaktor  
092 05 05 för kodaddering  
093 07 7 Övre registergräns  
094 42 STD för tider  
095 04 04  
096 05 5  
097 02 2  
098 42 STD Pekregister för alfa-  
099 06 06 numerisk kodlagring  
100 25 CLR  
101 91 R/S

102 76 LBL  
103 12 B  
104 43 RCL Övre minnesgräns  
105 04 04  
106 42 STD  
107 02 02  
108 75 -  
109 07 7  
110 95 =  
111 42 STD Antal tider  
112 00 00  
113 69 DP  
114 32 32  
115 73 RC+  
116 02 02  
117 65 x  
118 43 RCL  
119 99 99 Mult. faktor  
120 22 INV  
121 28 LDG  
122 95 =  
123 22 INV  
124 59 INT

125 65 x  
126 01 1  
127 00 0  
128 00 0  
129 85 +  
130 93 =  
131 05 5  
132 95 =  
133 59 INT Register kod  
134 42 STD  
135 01 01  
136 73 RC+  
137 01 01  
138 69 DP  
139 04 04  
140 73 RC+  
141 02 02  
142 75 -  
143 43 RCL  
144 01 01  
145 65 x  
146 43 RCL  
147 05 05  
148 95 =  
149 69 DP  
150 06 06  
151 97 DSZ  
152 00 00  
153 01 01  
154 13 13  
155 25 CLR  
156 98 RDV  
157 93 RDV  
158 98 RDV  
159 91 R/S

058 76 LBL  
059 11 A  
060 95 +  
061 43 RCL  
062 06 06  
063 65 x  
064 43 RCL  
065 05 05  
066 95 =  
067 42 STD  
068 00 00  
069 32 XIT  
070 72 ST+  
071 06 06  
072 69 DP  
073 26 26  
074 43 RCL  
075 04 04  
076 42 STD  
077 02 02  
078 06 06  
079 81 RST

### Registerlistning

0. 00  
55. 01  
7. 02  
15. 03  
0.00001 05  
60. 06  
1.388755 07  
1.383359 08  
1.382157 09  
1.381954 10  
1.381456 11  
1.380253 12  
1.373558 13  
1.372652 14  
0. 15

0. 50  
0. 51  
36371731. 52  
30132335. 53  
22353236. 54  
37233217. 55  
33323313. 56  
2172416. 57  
31174135. 58  
36373513. 59  
0. 60

000 42 STD  
001 01 01  
002 85 +  
003 01 1  
004 95 =  
005 32 XIT  
006 43 RCL  
007 02 02  
008 69 DP  
009 00 00  
010 39 39  
011 85 +  
012 43 RCL  
013 01 01  
014 95 =  
015 95 +  
016 02 2  
017 95 =  
018 59 INT  
019 42 STD  
020 03 03  
021 43 RCL  
022 00 00  
023 32 XIT  
024 73 RC+  
025 03 03  
026 77 GE  
027 00 00  
028 36 36  
029 43 RCL  
030 03 03  
031 42 STD  
032 02 02  
033 43 RCL  
034 01 01  
035 81 RST  
036 43 RCL  
037 05 03  
038 81 RST

039 43 RCL  
040 04 04  
041 75  
042 43 RCL  
043 01 01  
044 95 =  
045 48 ENO  
046 00 00  
047 69 DP  
048 21 21  
049 63 EN+  
050 01 01  
051 97 DSZ  
052 00 00  
053 00 00  
054 47 47  
055 69 DP  
056 24 24  
057 91 R/S

### Använda labels

059 11 A  
081 15 E  
103 12 B

### Mjukvaruspecifikation

Typ av räkare: TI-58/59 + PC-100 A,B el. C  
Exekveringstid ca 30 s. per tid.  
Uppdelning 10 OP 17. I kortsida  
160 programsteg

STEP	PROCEDURE	ENTER	PRESS	DISPLAY	COMMENTS
1	Initiering; Ange antal decimaler	m	E	0	
2	Alfa. kod enligt instruktionsbok	kkkkkkk	xat	0.	
	Tid	m.ssss	A		
3	Upprepa enl. Step 2 För varje tid.				
4	Utskrift av res. lista		B		Utskrift av resultatlista
	Exempel Slalom Initiera med 4 decimaler	4	E	0	
	Alfa. kod för Enn	173131	xat		
	Tid för Hans Enn	1.2111	A		
	Osv				
	Utskrift efter 3 åkare		B		1.2111 EHN
	Lagra fler tider enl. ovan				1.2446 ADGR
	Utskrift		B		1.2503 RHYN
					1.2111 EHN
					1.2116 KRIZ
					1.2129 STEN
					1.2168 STRE
					1.2184 WENZ
					1.2223 HEID
					1.226 BURG

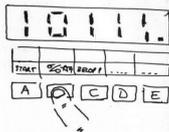


## Sortera efter storlek

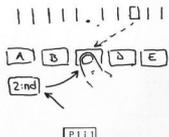
Att sortera siffervärden i storleksordning är ganska tidsödande på en programmerbar räknare. I matematikmodulen finns ett program MU-06 där själva sorteringsrutinen upptar 100 programsteg. Här visas ett enklare program på 40 programsteg som i gengäld är långsammare. Att sortera 30 värden tar med detta drygt sex minuter medan MU-06 gör det på ganska precis fyra, och det är vänta att skillnaden blir större med fler värden att sortera. (Med 15 värden tar det ca 1/2 minut för både.) Listningen nedan klarar 57 värden och skall ligga i minnen 01 - 57. Om "59" ändras till "99" och "58" till "98" går det med 97 värden (uppdelning "10 op 17"). Programmet startas med antal värden följt av A.

000	76	LBL	021	77	GE
001	11	R	022	00	00
002	75	-	023	32	32
003	01	=	024	69	DP
004	95	=	025	30	30
005	42	STD	026	63	EX+
006	59	59	027	00	00
007	01	I	028	69	DP
008	42	STD	029	20	20
009	00	00	030	72	ST+
010	43	RCL	031	00	00
011	59	59	032	97	D52
012	42	STD	033	58	58
013	58	58	034	00	00
014	73	RC+	035	16	16
015	00	00	036	97	D52
016	32	X:Y	037	59	59
017	69	DP	038	00	00
018	20	20	039	97	DP
019	73	RC+	040	91	R/S
020	00	00			

## INSTRUKTIONS- STYRT PROGRAM- LITAN SKRIVARE!



- Välj A som "startknapp".
- Avsluta initieringsrutinen i A med att mata ut ettorsamt en 0 (nolla) i den position som motsvarar tangent A-E enligt figur.
- Lämpig text på ett kort insett ovanför A-E ger instruktion om vad som skall matas in....
- För A-E kan förlagsvis indikationen ligga efter ett komma.



### Programavslutning - ibland

**LBL** Vid ettens sekvens är inte så ovanlig. Om t ex detta pgm skall upprepas ett antal gånger i följd med olika indata, så skall man alltså första gången trycka A och sedan två gånger R/S efter de tre respektive inmatningarna.  
**R/S** Men åter gången skall man trycka A igen. Då är det väldigt lätt att glömma vixlingen till A var 4:e gång.  
**P23j**

### Variabelnummer i display

**LBL** Betrakta vid ettens sekvens. Avsikten här är tydligen, att display skall visa variabelnumret nr i tur och ordning, så att man vet vad som skall inmatas.  
**R/S** Men när i står i display, och man vill byta ut ettan mot t ex 3,21 så får man 13,21, om man inte först gör CE, vilket är bevärligt och lätt att glömma - med förordande verkan.  
**R/S** **Tips:** Lägg in ett likhetstecken (=) mellan variabelnumret och R/S. Eller - om = står vilande operatorer - "innes" variabelnumret med parenteser, t ex  
**LBL A ... (1) R/S ... (2) R/S etc.**  
**P23j**

### Förberedelse till numerisk adressering

När man gör ett pgm, som innehåller hoppinstruktioner, så använder man oftast LBL-adresser vid första inmatningen. Senare vill man kanske gå över till NUM-ADR. Om man då med IRS gör plats för dessa extra etta, så förändras hela stegkolan, och man får bevärligt med att extra rutt stegen i NUM-ADR.  
**Tips:** Läg in ett NOP från början efter varje LBL-ADR, så finns platsen redan där, när den senare behövs. Ex. GTO LMX NOP andras lätt till t ex GTO 03 26. Själva labeln LMX får ligga kvar (för förutsätter att pgm-stegen ricker). Genom att göra RST OP 08 får man ju LBL-listan, och om där står:  
 325 23 LMX, så vet man att GTO LMX NOP skall ändras som ovan.  
 Efter eventuella framtida pgm-ändringar är det lätt att med en ny LBL-lista finna de nya NUM-ADR som man skall ändra till i hoppinstruktionerna.  
**P23j**

### SLUT PÅ LABELS?

Utöver de labels som visas bl a på den programmeringsblankett som kommer tillsammans med räknaren går det faktiskt att använda även ett antal "sammansatta" koder. Knepet är egentligen bara att programmera. De koder som går att använda är 62, 63, 64, 72, 73, 74, 82, 83, 84 och 92. Programmeringen görs lämpligen med "STO 62 BST Lbl SST" och motsvarande för t ex "GTO PGK" eller "xet PGK"  
**P11j**

### Noggrannhet hos INV log och x<sup>x</sup>

Varken INV log eller x<sup>x</sup> är exakta, vilket kan vara bra att komma ihåg t ex vid tester med "xet" eller liknande. Det är givetvis lätt att köra ett program för att lista skillnaden till det exakta värdet, beräknat med EE resp t ex 2 x 2 x ... Nedan visas "feltabeln" för 'x INV log' och 2<sup>x</sup>.

x INV log - I EE x	2 <sup>x</sup> - 2'2'2'2' (x st)		
0,	0,	0,	01
0,	0,	0,	02
0,	01	1,-12	03
0,	04	-1,-11	04
0,	05	0,	05
0,	06	05	1,-11
0,	07	0,	07
-0,000005	05	0,000003001	08
-0,00009	07	0,000003003	09
-0,0013	08	0,	10
-0,017	09	0,000000001	11
0,	10	0,000000002	12
-0,5	11	0,000000002	13
-3,	12	0,000000002	14
-30,	13	0,000000005	15
-700,	14	0,00000001	16
-41000,	15	0,00000002	17
-410000,	16	0,00000006	18
-4100000,	17	0,0000001	19
-53000000,	18	0,0000002	20
-570000000,	19	0,0000005	21
0,	20	0,000001	22
-5000000000,	21	0,000002	23
-9,	10	0,000006	24
-3,	11	0,000013	25
-7,	12	0,00001	26
-8,	14	0,00004	27
-8,5	15	0,00008	28
-8,9	16	0,0002	29
-9,3	17	0,0004	30
-9,7	18	0,0009	31
0,	30	0,001	30
-9,5	19	0,003	31
-9,	20	0,009	32
-9,	21	0,016	33
-7,	22	0,03	34
-1,21	25	0,06	35
-1,	26	0,16	36
-1,29	27	0,3	37
-1,33	28	0,7	38
-1,37	29	1,4	39
0,	40	2,	40
0,	29	4,	41
-9,	30	4,2	42
-3,	31	8,	43
-3,	32	16,	44
-3,31	35	120,	45
-2,25	36	240,	46
-2,19	37	400,	47
-2,13	38	500,	48
-2,07	39	2100,	49
0,	50	5000,	50

### x MODULO t

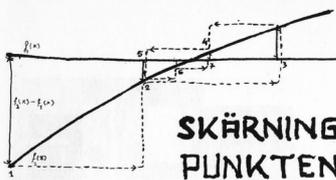
För att beräkna x modulo t - dvs resten då x divideras med t - brukar jag använda en rutin enligt denna remsa. Den anropas genom:

```
t x t x A
```

00	76	LEL	06	65	x	12	59	INT
01	11	R	07	53	(	13	54	)
02	53	(	08	32	X:Y	14	92	RTH
03	24	CE	09					
04	75	-	10	32	X:Y			
05	32	X:Y	11	54	)			







# SKÄRNINGSPUNKTEN

## MELLAN TVÅ EKVATIONER I 16 STEG

Programmet beräknar avståndet mellan två kurvor och jämför om avståndet mellan kurvorna ökat eller minskat efter en "framstegning".

Så länge avståndet minskar fortsätter programmet i samma riktning, men om avståndet börjar öka "vänder" programmet och "går" i motsatt riktning (efter x-axeln) med hälften så långa "steg".

Liknar den numeriska intervallhalveringsmetoden, men SE UPF1 För tredjegradsfunktioner och ex vis avstånd" som du får fram!

DIAGRAMMA DI FLESSIONOTE	TASTO	POS	CONDIC	COMENTARIO
FLUJEDIAGRAMNOTATER	TAST	TRIN	KODE	KOMENTAR
STRUKTURSIEMIAOPISERENGEN	LOCS	POS	KODE	KOMENTAR
RCL 2	00	33	2	
SUM 1	01	34	1	
SBR 1	02	61	1	
STO 3	03	32	3	
STO 3	04	61	2	
SUB 3	05	34	3	
RCL 3	06	33	3	
Pause	07	36		
IXI	08	42		
XXE	09	42		
X > E	10	76		
RST	11	71		
2	12	2		
13	84			
INV PRD 2	14	39	2	
RSP	15	71		
Lbl 1	16	86	1	f(x)
RCL 1	17	33	1	
X	18	23		X
-	19	65		Y
2	20	2		Y
X	21	55		X
RCL 1	22	33	1	
+	23	75		B
1	24	7		
+	25	85		
INV SBR	26	61		
Lbl 2	27	86	2	f(x)
RCL 1	28	33	1	
IN X	29	13		Y
+	30	75		X
5	31	5		B
-	32	85		Y
INV SBR	33	61		
34				
35				

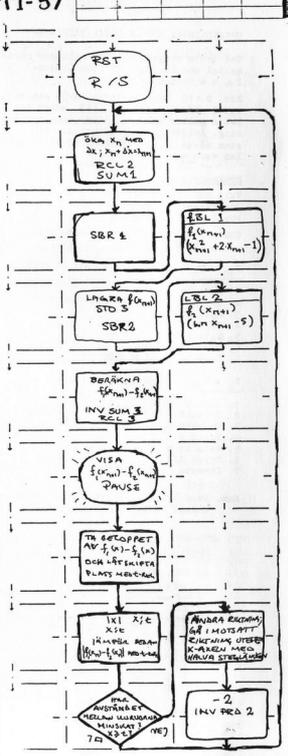
  

REGISTRI DATI	ETICHETTE
DATA REGISTER	LABELS (OO)
DATAREGISTERS	ETICHETTE
DATAREGISTERS	LABELS
0	n 0
1	X <sub>1</sub> 5V 1 f <sub>1</sub> (x)
2	Δx 5V 2 f <sub>2</sub> (x)
3	f <sub>1</sub> (x <sub>1</sub> ) - f <sub>2</sub> (x <sub>1</sub> ) 5X 3
4	Δx <sup>2</sup> 4
5	(ΔOS) 5Xy 5
6	(ΔOS) 6
7	f <sub>1</sub> (x <sub>1</sub> ) - f <sub>2</sub> (x <sub>1</sub> ) + 1 7
8	TEXAS INSTRUMENTS 8
9	9

BÖRJA MED ATT ANGE STARTVÄRDE (STO 1) OCH STEGLÄNGD (STO 2)!

PROGRAMBITEN 79-3/4

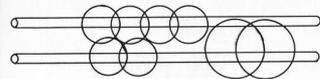
# TI-59 INTERVALHALVERINGSMETODEN



- CLR Efter ca 30 sek.....
- STD 2
- 0.002763
- STD 1 -0.5757703
- 1 -0.2933982
- 0.002763
- RST 0.2894937
- IR/S 0.1455746
- 0.002763
- 0.1490677
- 0.0761895
- 0.002763
- 0.0711518
- 0.0941309
- 0.002763

AVLÄS X I REG 1 3.5

# ÖVERHÖRNING MELLAN LEDARE



För att beräkna L matar man in ledarnas längd, l, och avståndet dem emellan, d. Den inducerade spänningen får man efter att ha matat in strömbändring, di, och per tidsenhet, dt. Observera att vid spänningsberäkningen används värdet för den gemensamma induktansen, varför den alltid först måste beräknas.

## ÖVERHÖRNING MELLAN LEDARE.

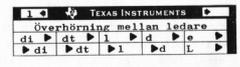
Induktansen mellan ett par parallella ledare gör att en spänning induceras i den ena ledaren när strömmen varierar i den andra. Detta fenomen är ett rätt vanligt problem i elektronikutrustningar. Ibland kan man till och med medvetet använda sig av detta fenomen. Ett exempel på det är när man lägger in (små) induktansar direkt, med foliens hjälp, på ett kretskort.

I vilket fall som helst så underlättar detta program väsentligt beräkningsarbetet. Det är baserat på Rost's formel för induktans mellan ledare.

Programmet beräknar induktansen mellan två ledare eller den inducerade spänningen utifrån variablerna:

- Lbl:
- A: Enter di = strömbändringen... (A)
  - A': Recall di
  - B: Enter dt = ... under tiden t. (sek)
  - B': Recall dt
  - C: Enter l = ledarnas längd (cm)
  - C': Recall l
  - D: Enter d = avstånd mellan ledarna (cm)
  - D': Recall d
  - E: Beräkna L (induktansen) (µH)
  - E': Beräkna e (inducerade spänningen) (V)

000	76	LBL	041	19	D*	072	16	A*
001	16	A*	042	38	R*	073	55	+
002	43	RCL	043	54	)	074	17	B*
003	01	01	044	34	FX	075	55	+
004	92	RTN	045	23	LHX	076	75	-
005	76	LBL	046	07	077	52	E	
006	17	B*	047	53	(	078	66	E
007	43	RCL	048	01	1	079	95	+
008	02	02	049	85	+	080	22	IHV
009	92	RTN	050	19	D*	081	52	E
010	76	LBL	051	33	X*	082	91	R/S
011	18	C*	052	55	+	083	76	LBL
012	43	RCL	053	18	C*	084	11	H
013	03	03	054	33	X*	085	42	STD
014	92	RTN	055	54	)	086	01	01
015	76	LBL	056	34	FX	087	91	R/S
016	19	D*	057	85	+	088	76	LBL
017	43	RCL	058	19	D*	089	12	B
018	04	04	059	55	+	090	42	STD
019	92	RTN	060	05	091	02	02	
020	76	LBL	061	54	)	092	91	R/S
021	15	E	062	95	+	093	76	LBL
022	25	CLR	063	42	STD	094	13	C
023	93	+	064	05	095	42	STD	
024	00	0	065	91	R/S	096	03	03
025	02	2	066	76	LBL	097	71	R/S
026	65	X	067	10	E*	098	76	LBL
027	18	C*	068	25	CLR	099	14	D
028	65	X	069	43	RCL	100	42	STD
029	53	(	070	05	101	04	04	
030	53	(	071	65	X	102	91	R/S
031	18	C*						
032	55	+						
033	19	D*						
034	85	+						
035	53	(						
036	01	1						
037	85	+						
038	18	C*						
039	33	X*						
040	55	+						

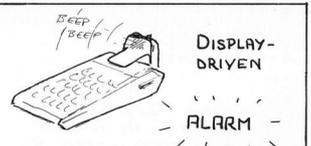


## Roligt, snabbt, rätt!

Vårt program för Texas TI-59 täcker de flesta av byggkonstruktörens vardagsproblem och kostar inte mer än att göra motsvarande beräkning för hand några gånger. Kontinuerliga balkar, pelardäck, excentriskt tryckta betong- och stålplåtar, betongvärmis, sågvarvning, ramar, takstolar, stödmurar, pålgurpper. Och så säljer vi TI-59 med skrivare och Texas "modul" med färdiga program. Ring oss så kommer vi och visar, eller också skickar vi prislista eller program till påseende.

Carl-Adolf Granholm  
Box 5030  
42105 V. Frölunda  
tel 031-292075

Lars Hedlund  
Årstavägen 27  
12168 Johanneshov  
tel 08-913897



Hög alarmsignal ljuder när displayen fylls med siffror.

Kan ex vis användas till att ljudligt indikera när din TI-59a stannat för ny datainmatning, programfel, programslut etc.

Timerprogram medföljer (TI-57/58/59)

MANIAC  
Ormingår 14 B  
132 00 SALTSJÖ-BOO  
postgiron 46 67 66 - 3

285:-  
plus moms  
och frakt.

PROGRAM FRÅN  
NILS ANDREAS ERSTAD  
NORGE

DATA -LAGRING, -GJENKALLING OG -INNHOLDSTESTING

TI-59 Q103  
Lagrer og gjenkaller data - søksti enheter på magnetkort.  
Hver enhet kan ha ti variabler + identifikasjonssifre; hver variabel 10 verdier, 1 eller 2 variabler med ett eller flere sifre og tilhørende betingelser testes for: (a og b); (a eller b); (ikke a men b). En variabel testes alene, eller utskrift av en variabel med beregning av  $\bar{x}$  og  $\bar{X}$ . Minneoppdeling av maskinen. Benyttes til bearbeiding av data fra spørreskjema o.l., eller til enkle katalogiseringsformål.  
(Skriver) Tid: 2-4 sek./støtnehet Sider 1,2 Støt: 340  
(Data 3,4)

## DATALAGRING INNEHÅLLSTESTNING

### STATISTIK

PRESENTASJON AV FORFATTEREN

Oppdragsforsker for KUD - "Utdanning hos psykiatriske pasienter". Rektor i skole for slik undervisning. Cand. paed.  
Nils Andreas Erstad, Molde, Norge.

Som samfunnsforsker (pedagog) har jeg nå og da behov for statistiske beregninger. Etter å hatt stor glede av en Casio FX-19, falt valget på TI-59 med skrive-enhet. Valget av denne i stedet for den eneste konkurrenten, ble gjort dels på grunn av statistikkmodulen, og dels fordi AOS-systemet ser enklere ut enn RPN for en ikke-matematiker.

For samfunnsforskeren er det særlig to begrensninger som viser seg ved bruken av dette utstyret:

Det første er å bli tatt alvorlig når en innrømmer at statistikk ikke er resultat av en imponerende stor og uforståelig datamaskin, men er gjort på hjemmekontoret med den lille maskinen en har i vesken.

Det andre problemet er å overføre data fra spørre/opplysningskjema til former som lar seg behandle statistisk.

Det første problemet ligger utenfor denne artikkelens område, og lar seg vel best løse ved at flere samfunnsforskere får se hvor anvendelig TI-59 er.

Det andre problemet er nedenfor stående program en mulig løsning på. Det gjør det mulig å bruke maskinen som en "hullkortsorterer" med utskrifter og endel logiske tests.

#### PROGRAM-PROBLEM

Det er tatt særlig hensyn til rimelig enkelhet i bruk av programmet, og til at gjennomkjøring bør kunne skje på rimelig tid. Det gir begrensninger i antallet tests - to er valgt.

Dette gir begrensninger i informasjonsmengden. Men i selve opplegget ligger allerede slike begrensninger. Mens et hullkort gir plass for 80 x 10 (variabler x verdier), og via en EDB-maskin kan kjedes sammen til lengere variabelkjeder, får vi max. 10 x 10. (Figur 1). I tillegg får vi plass til identifikasjonssifre utenom de 10 variablene.

Fig. 1. - "hullkort" for TI-59

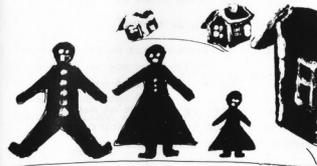
Variable nr:	Minne nr.
10	9
9	8
8	7
7	6
6	5
5	4
4	3
3	2
2	1
1	0
0	0

Pilk har i nr. 78-2 beskrevet hvordan 13 sifrer lagres og gjenkalles. Her er valgt å benytte 10, fordi dette gir enkel adgang til utskrift, og forenkler alle rutiner. Dette bidrar også til å begrense programmets lengde, og tid for gjennomføring. Særlig viktig er det å gjøre innlesingsprosedyren så enkel at kontorpersonele kan overta dette.

#### BEGRENSNINGER

Å ha begrenset plass til variabler gjør selvdisiplin nødvendig. Dette kan være positivt, ved at kravene til hypoteser og begrunnet utvalg av variabler/verdier øker. Behovet for å bestemme hvilke variabler som skal ligge nær hverandre i sifferserien stiller krav til skikkelige hypoteser. Muligheten for å undersøke en rekke tilfeldige variabler for å se om det ikke skulle dukke opp en eller annen signifikans er liten.

Samtidig gir metoden god kontroll og kontakt med den statistiske bearbeidingen, og et mye bedre kjennskap til data.



FLER ENN 10 VARIABLER

Ved å kombinere variabler med få verdier, kan antallet variabler økes noe. En har bruk for opplysninger om kjønn, har dette bare to verdier. Om viktigste bostedsklasse går mellom by og landkommune, har dette også to verdier. I stedet for å benytte to variabelplasser (siffer), kan kvinne i by få verdien 0, kvinne i landkommune 1, mann i by 2, og mann i landkommune 4. Som en ser, kunne også en tredje to-verdi-variabel vært kombinert her.

For en går for langt i å kombinere variabler, skal en være klar over at det øker arbeidet med å plukke ut opplysninger fra data. Dessuten vanskeliggjør det overføring fra spørreskjema til tallform, og dermed øker sjansene for feil.

#### HENSINT OG FUNKSJONER

Programmet er konstruert for å lette arbeidet med å overføre data til tabeller og diagrammer - og dermed til tallform som kan danne grunnlag for statistiske beregninger - enten en nå benytter standardmodul, statistikkmodul, eller egne programmer.

Data behandles på fem måter. Det er en ren utskriftsfunksjon (med sum og middelverdi), og fire "hvis"-funksjoner. Disse presenteres i tabell 1.

#### TABELL 1.

Press	Funksjon	Flagg
(2nd)	(a og b)	0 <sup>1)</sup>
(A')	Tester om en variabel tilfredsstillen en spesifisert betingelse, og en annen variabel tilfredsstillen sin spesifiserte betingelse.	
(2nd)	(a eller b)	1 <sup>1)</sup>
(B')	Tester om en variabel tilfredsstillen en spesifisert betingelse, eller en annen (eller samme) variabel tilfredsstillen en annen betingelse.	
(2nd)	(ikke a men b)	2 <sup>1)</sup>
(C')	Tester om en variabel er ulik spesifisert krav i en variabel, men tilfredsstillen spesifisert krav i en annen variabel.	
(2nd)	(hvis a)	3 <sup>1)</sup>
(D')	Tester om en variabel tilfredsstillen en betingelse.	
(D)	Gir utskrift av en variabel for alle dataenheter, samt sum av variabelverdier ( $\Sigma x$ ) og middelverdi ( $\bar{x}$ ).	4 <sup>1)</sup>

1) En variabel kan bestå av flere siffer, og dermed også av en variabelkombinasjon hvis variablene ligger ved siden av hverandre.  
i) Utskrift: x FLAGG.

#### INITIALISERING: (2nd)(E')

Innlesing av programkort skjer ved normal oppdeling av minnene. Etter innlesing av program benyttes initialiseringsprosedyren (2nd)(E'). Her gis identifikasjon av programmet ved utskrift: VARIABLE-DIGIT EQ.

Maskinen stilles om til 70 minner/400 programsteg. Alle minner tømnes - både data og mellomregnings/styringsminner. b-register tømnes. "FLAG" lagres i Op 04 for senere benyttelse.

Initialiseringsprosedyren benyttes også mellom hver ny innlesing av data, enten dette skjer direkte, eller fra kort.

## INNLESING AV DATA (C)

Innlesing av data skjer første gang alltid fra tangentbord: inntil 10-sifrede tallrekker slås inn, og lagres ved (C). Første dataenhet lagres i minne 00, 2.de i 01 osv. Så lenge vi nummererer i overensstemmelse med dette, vil altså maskinen selv sørge for rett identifikasjonsnummer. Hvis hvert datakort får bokstav, kan vi få identifikasjonsverdiene 00 - 59, A - Å. Datakortenes kode avmerkes manuelt på utskriftene.

Maskinen gir feilindikering ved forsøk på å slå inn data i minne 60. Dette forstyrrer ikke videre behandling, og oppheves med (CLR) før innlesing på kort, side 3 og 4.

Det er ingen feilrettingsprosedyrer. Det er i alle fall nødvendig med nøyaktig kontrolllesing av data fra utskriften som maskinen gir etter hvert som data leses inn. En mal laget av en tom minnutekstrif letter identifikasjon av minner. Ved feil. Feil en vet er gjort kan markeres på utskriften etter hvert.

Display gir beskjed om nye dataminnes nr. (Forutsatt at ikke "RESET" (E) er benyttet i stedet for (2nd)(E)'): Da vil forrige minnes nummer vises.)

Retting skjer enten under innlesing, eller etterpå ved kontroll, ved minne nr. xy slik: (data slås inn - max. 10 siffer)(STO) xy.

På grunn av at maskinen stopper eksekvering ved første dataminne som er tomt, må ingen mellomliggende dataenheter være satt lik 0. Om det er ønskelig å markere manglende svar, kan dette gjøres ved å la det n'ite siffer få en verdi som ikke er benyttet. På denne måten er det også lett å finne fram hvor mange ubesvarte spørreskjemaer (e.a.) som finnes, uavt å oppta en egen variabel til dette.

## LAGRING AV DATA PÅ KORT

Etter at data er lest inn, må ikke initialisering foretas før data er overført til kort, eller er uten videre interesse.

Kortene leses på vanlig måte inn, side 3 og 4. Ved flere kort, markeres dette manuelt, f.eks. ved bokstavkoder. Ved å lese inn kortene med inndeling 399.69 vil bare programmer med denne inndeling akseptere kortene, og det vil ikke være nødvendig å endre oppdeling i dette programmet etter initialisering.

## RESET-FUNKSJONEN (E)

Mellom datainnlesing (manuelt eller fra kort) og behandling av data, og mellom hver behandling av data, benyttes resetfunksjonen (E).

Med denne ~~tømmes~~ alle mellomregningsminner, t-register, og alle flagg fjernes. Telleregister får rett informasjon. Dataregister berøres ikke.

Hvis ikke (E) benyttes før ny behandling, inkludert den første, vil minne 00 bli hoppet over.

## VALG AV TESTBETINGELSE

Se tabell 1 for valg av testbet.

## TESTBETINGELSER (B) (R/S) ...

Ved logiske tester og utskrift må maskinen også få beskjed om hvilke variabler og betingelser som skal tilfredstilles.

Plassering av første variabls siste siffer (nummerering: se fig. 1.) slås inn, og lagres i formen  $10^{x-1}$  ved (B). Antall siffer i første variabel mates inn med (R/S), og lagres som  $10^x$ . Utskriften viser alltid det innslåtte tall - display den lagrede verdien.

Hvis maskinen nå er stilt inn for ren utskrift, skrives det ut 0 for de andre mulige verdiene.

Hvis ikke, slås sammenligningsverdi inn med (R/S).

Hvis maskinen er stilt inn på enkelt-sammenligning, skrives 0 ut for de siste mulige verdiene. Hvis ikke gjentas sekvensen for andre variabel, men nå med (R/S) etter alle verdiene, også plassering. Se ellers diagram 1.

## FEIL VED BRUK AV (B)

Ved feil bruk av (B) (R/S) i den foregående sekvensen, hender det at oppdelingen endres. Dette kan kontrolleres slik: (2nd)(Op)16 - display skal da vise 399.69. Ved feil vil et annet tall vises (159.999). For ikke å miste data, rettes feilen opp slik: 7 (2nd)(Op) 17. Display: 399.69.

## EKSEKVERING (A)

Etter at data, funksjonsmåte og betingelser er matet inn, startes eksekvering ved (A). Eksekveringstid er avhengig av data: Mengde, hvor mange som tilfredstiller betingelsene, og funksjonsmåte - anslagsvis mellom 2 og 4 sek. for hver dataenhet.

For alle funksjonsmåtene gis det utskrift av alle dataenheter som tilfredstiller betingelsene, med identifikasjonsnummer på egen linje. (Unntatt ved ren utskriftsfunksjon.) Som avslutning skrives antallet enheter som tilfredstiller betingelsene. Ved utskrift fås også summen av alle verdier (Σx), og middelverdi (x̄). Merk at middelverdien blir feil her om det finnes tomme dataminne som ikke skal telles med!

Foreta alltid en rask kontroll av at funksjonsmåten er som bestemt! Feil bruk av (E) er en god måte å kunne galt ut på! Se ellers diagram 2.

## KORT REPETISJON AV VIKTIGE PUNKTER

Rekkefølgen er:

a) Innmatning av data - (2nd)(E)'.  
data (D) data (D) osv., eller: (2nd)(E)' kjøring av kort s. 3 og 4.

b) Behandling av data - (E), velg funksjon, velg betingelser, (A). Dette gjelder både ved første kjøring, og ved seinere.

## KORT OM PROGRAMKONSTRUKSJONEN

På enkelte punkter er valgt mer primitive løsninger enn hva som ellers kunne vært mulig.

I stedet for Dsz, benyttes 1 SUM - delvis fordi jeg på det tidspunktet ikke kjente til hvordan en kunne benytte minnene over 0 - 09. Men det har også gjort det mulig å benytte minne 00, og foreta oppvertelling, noe som forenkler overakten i data senere. Dette forholdet gjør også at det ligger endel kontrollor ved set i programmet som kanhende kunne vært unngått.

Forøvrig benytter programmet flere x=t tester og flaggtester for styring og kontroll.

I størst mulig utstrekning er labels enten plassert i begynnelsen av programmet, kombinert med GTO-instruksjoner, eller erstatter med rene GTO-instruksjoner/direkte adressering. Dette øker hastigheten på et ellers langsomt program.

Under utarbeidelse og utprøving ble bruken av direkte adressering lettet ved at mange NOP-instruksjoner var lagt inn. Slik kunne hver ny programsekvens begynne på xx0 eller xx5,te steg, og endringer foretas uten at forskyvninger oppsto. Først når programmet virket etter forutsetningene ble disse fjernet, og GTO-adressene korrigert.

To labels er beholdt etter at korrigeringsarbeidet var over, selv om de er uten funksjon: (EE) og (V). Dette fordi det da vil være lettere å foreta senere endringer av programmet.

Det benyttes også en ikke-definert label:(lnx).

Denne benyttes til å gi feilindikering ved forsøk på å slå inn for mange data.

## EKSEMPEL PÅ BRUK

La oss tenke oss en undersøkelse der det bla. er spurt om følgende variabler, med tilhørende verdier og variablenes plassering i sifferrekken:

Kjønn: Mann (1) kvinne(2)	Variabel 10
Bosted: By (1) landkommune (2)	Variabel 9
Ektesk. st. Ugift (1) gift (2)	Variabel 8
tidl. gift (3)	
Barn: Barnløs (1) har barn (2)	variabel 7
Alder: to siffer	var. 5 og 6
Andre variabler (x)(her: 0)	var. 1 - 4

5 personer, med tilhørende innhold i data som leses inn i maskinen, (identifikasjonsnummer i parentes foran - skal ikke slås inn!):

Mann; by; ugift; 1 barn; 43 år.	(00) 1112430000
Mann; by; gift; 3 barn; 50 år.	(01) 1122500000
Kvinne; landkomm.; gift; 2 barn; 23 år.	(02) 2222300000
Kvinne; by; tidl. gift; 0 barn; 39 år.	(03) 2131390000
Mann; landkommune; gift; 0 barn; 21 år.	(04) 1221210000

Ved å følge prosedyrer for initialisering og innlesing av data, har vi fått denne utskriften:

VARIBLE-DIGIT EQ	1112430000.	00
	1122500000.	01
1124300000.	2222300000.	02
1122500000.	2131390000.	03
2222301111.	1221210000.	04
1221210000.	0.	05

Vi ser at første utskrift inneholdt to feil som er rettet, og kontrollert i utskriften til høyre (CLR)(INV)(2nd)(List).

III

Menn i by med barn, og gifte.  
 Framgangsmåte:  
 (E) (2nd)(A') 9 (B) 2 (R/S) 11 (R/S)  
 8 (R/S) 1 (R/S) 2 (R/S) (A):  
 0. FLAG

UTSKRIFTER:

I  
 Oversikt over alder (variabel 5 og 6).  
 Framgangsmåte: (E)(D) 5 (B) 2 (R/S)(A):

4. FLAG (= funksjon (D))  
 5. (siste siffer nr.)  
 2. (antall siffer)  
 0. (skrives automatisk ut)

0. =  
 1. =  
 0. =  
 43. } (utskrift av siffer 5 og 6)  
 50. }  
 23. }  
 39. }  
 21. }

5. (antall dataenheter)  
 176. (sum av alles alder)  
 35.2 (gjennomsnittsalder)

1122500000.

1. 22222300000.  
 2. 21313900000.  
 3. 12212100000.  
 4.

II  
 Alle som har barn. Framgangsmåte:  
 (E)(2nd)(D') 7 (B) 1 (R/S) 2 (R/S)(A):

3. FLAG (= Funksjon (D'))

7. (siste s. nr.)  
 1. (ant. s.)  
 2. (x=et betingelse)  
 0. (skrives aut.)  
 0. =  
 0. =

1112430000. (data ved x=et)  
 0. (identifikasjon)  
 1122500000.  
 22222300000.

2. FLAG  
 3. (Sum ved x=et)

IV  
 Gift eller tidligere gift. Framgangsmåte:  
 (B) (2nd)(B') 8 (B) 1 (R/S) 2 (R/S)  
 8 (R/S) 1 (R/S) 3 (R/S) (A):  
 1. FLAG (se tidl. utskr.)  
 8.  
 1.  
 2.

1122500000.  
 1. 22222300000.  
 2. 21313900000.  
 3. 12212100000.  
 4.

2131390000.

3. 12212100000.  
 4.  
 2.

Ti 58/59

TITLE : DATA -LAGRING, -GJENKALLING OG -INNHOLDSTESTING

PAGE : OF :

STEP	PROCEDURE	ENTER	PRESS	DISPLAY	COMMENTS
	Innlesing av kortside 1 og 2				
1	Initialisering, fbr innspill / lesing av data		2nd E	0	Med PC-100: VARIABLE-DIGIT EQ
2	Innlesing av data	data 1 000 000 000	C (for het)	en- neste data nr	Med PC-100:Utskrift som display Blinkende display ved forsok på å lese inn mer enn 60 data
2b	Lesing av datakort	side nr	-	side nr	
3	Klargjøring for test		E	0	
4	Valg av logisk beting.		2nd A 2nd B 2nd C 2nd D	" " " "	Med PC-100: x Flag " " " "
5	Betingelser:				
	a) Variabels siste siffer	x	B	10 <sup>x-1</sup>	Med PC-100:Utskrift som display
	b) Variabel - lengde	y	R/S	10 <sup>y</sup>	
	c) Betingelse	z	R/S	z	
	Hvis ikke D (Flag 4)				
	d) som a), 2.variabler	x	R/S	10 <sup>x-1</sup>	"
	e) som b), 2."	y	R/S	10 <sup>y</sup>	"
	f) som c), 2."	z	R/S	z	"
6	Kjøring		A	n i Flag bet.	Med PC-100:Utskrift med alle dataenheter som overholder betingelsene, samt minne-nr. Der- etter n (antall som overholder betingelsene)
7	Fbr nye betingelser steg 3 og 4, deretter 5 og 6.				
8	Fbr nye data: steg 1 - osv.				

000	91 R/S	0432	XIT	086	43 RCL	129 00 0	172	99 PRT	215	99 PRT	253	91 R/S	301 03 03
001	76 LBL	044 73 RC	087	64 64	130 32 XIT	173 42 STD	216	99 PRT	259	76 LBL	302 01 1		
002	35 1/X	045 69 69	088	34 )	131 43 RCL	174 63 63	217	98 RDV	260	10 E	303 07 7		
003	37 IFF	046 67 EQ	089	33 (	132 50 60	175 98 RDV	218	91 R/S	261	07 7	304 03 3		
004	00 00	047 33 YX	090	59 INT	133 67 EQ	176 37 IFF	219	76 LBL	262	69 DP	305 04 4		
005	00 00	048 53 (	091	55 =	134 23 LNX	177 03 03	220	16 R	263	17 17	306 00 0		
006	32 32	049 73 RC	092	43 RCL	135 43 RCL	178 02 02	221	86 STF	264	47 CHS	307 00 0		
007	37 IFF	050 69 69	093	65 65	136 76 LBL	179 13 13	222	00 00	265	29 CP	308 00 0		
008	02 02	051 55 +	094	54 )	137 15 E	180 91 R/S	223	00 0	266	04 4	309 00 0		
009	11 A	052 43 RCL	095	53 (	138 01 1	181 99 PRT	224	69 DP	267	02 2	310 00 0		
010	61 GTD	053 61 61	096	22 INV	139 94 +/-	182 75 -	225	06 268	01 1	311 00 0			
011	01 01	054 54 )	097	59 INT	140 42 STD	183 01 1	226	91 R/S	269	03 3	312 69 DP		
012	08 08	055 53 (	098	65 x	141 69 69	184 35 =	227	76 LBL	270	03 3	313 04 04		
013	92 RTN	056 59 INT	099	43 RCL	142 25 CLR	185 32 INV	228	17 B'	271	05 5	314 69 DP		
014	76 LBL	057 35 =	100	56 56	143 42 STD	186 76 LBL	229	04 STF	272	02 2	315 05 05		
015	45 YX	058 43 RCL	101	54 )	144 67 67	187 42 STD	230	01 01	273	04 4	316 69 DP		
016	99 PRT	059 62 62	102	32 XIT	145 42 STD	188 64 64	231	01 1	274	01 1	317 00 00		
017	44 SUM	060 54 )	103	64 RCL	146 68 68	189 91 R/S	232	69 DP	275	03 3	318 02 2		
018	68 68	061 33 (	104	36 36	147 98 RDV	190 99 PRT	233	06 06	276	69 DP	319 01 1		
019	01 1	062 22 INV	105	32 INV	148 81 RST	191 22 INV	234	91 R/S	277	01 01	320 02 2		
020	44 SUM	063 59 INT	106	67 EQ	149 91 R/S	192 28 LDG	235	76 LBL	278	01 1	321 07 7		
021	67 67	064 55 x	107	11 A	150 76 LBL	193 45 STD	236	18 C'	279	04 4	322 01 1		
022	11 GTD	065 43 RCL	108	73 RC*	151 12 12	194 65 65	237	86 STF	280	02 2	323 03 3		
023	11 A	066 62 62	109	69 69	152 98 RDV	195 91 R/S	238	02 02	281	07 7	324 02 2		
024	32 RTN	067 34 )	110	99 PRT	153 99 PRT	196 99 PRT	239	02 2	282	01 1	325 02 2		
025	76 LBL	068 37 IFF	111	43 RCL	154 75 =	197 42 STD	240	69 DP	283	07 7	326 69 DP		
026	33 YX	069 04 04	112	69 69	155 01 1	198 66 66	241	06 06	284	02 2	327 04 04		
027	98 RDV	070 45 YX	113	99 PRT	156 35 =	199 98 RDV	242	91 R/S	285	00 0	328 98 RDV		
028	43 RCL	071 32 XIT	114	01 1	157 22 INV	200 91 R/S	243	76 LBL	286	01 1	329 25 CLR		
029	67 67	072 43 RCL	115	44 SUM	158 29 LDG	201 76 LBL	244	19 D'	287	06 6	330 91 R/S		
030	99 PRT	073 63 63	116	67 67	159 42 STD	202 34 YX	245	86 STF	288	69 DP	331 43 RCL		
031	87 IFF	074 67 EQ	117	61 GTD	160 61 61	203 25 CLR	246	03 03	289	02 02	332 68 68		
032	04 04	075 35 1/X	118	11 A	161 91 R/S	204 99 PRT	247	03 3	290	02 2	333 99 PRT		
033	03 03	076 37 IFF	119	91 R/S	162 99 RDV	205 99 RDV	248	99 RDV	291	04 4	334 35 +		
034	31 31	077 00 00	120	76 LBL	163 22 INV	206 99 PRT	249	06 06	292	02 2	335 43 RCL		
035	98 RDV	078 11 A	121	13 C	164 28 LDG	207 99 PRT	250	91 R/S	293	02 2	336 67 67		
036	32 RTN	079 37 IFF	122	99 PRT	165 42 STD	208 99 PRT	251	76 LBL	294	02 2	337 95 +		
037	76 LBL	080 03 03	123	72 ST*	166 82 82	209 98 RDV	252	14 D	295	04 4	338 99 PRT		
038	11 A	081 11 A	124	69 69	167 87 IFF	210 91 R/S	253	86 STF	296	03 3	339 98 RDV		
039	01 1	082 53 (	125	01 1	168 04 04	211 76 LBL	254	04 04	297	07 7	340 91 R/S		
040	44 SUM	083 73 RC*	126	44 SUM	169 02 02	212 55 E	255	69 DP	298	00 0			
041	69 69	084 69 69	127	69 69	170 03 03	213 76 LBL	256	69 DP	299	00 0			
042	25 CLR	085 55 =	128	06 6	171 91 R/S	214 99 PRT	257	06 06	300	69 DP			

LABELS, MINNEINNHOOLD.

Labels (A) og (B) finnes forklart i diagrammene, sammen med (V<sup>2</sup>), (EE) (1/x) (x<sup>n</sup>) og (x<sup>2</sup>). Om den udefinerte (ln x) sist i \*kort om programkonstruksjonen\*.

Forøvrig finnes de andre anvenderdefinerte labels forklart nedenfor:

Lbl C 121  
Lagrer data i det minne hvis nummer fins i minne 69. Summerer for hver gang 1 til minne 69. Tester om innhold i minne 69 er mindre enn 60. Ved 60, gis en beskjed som ikke er definert (Lbl. Ln x), og dermed feilindikasjon. (Blinkende display.)

Lbl E 137  
Lagrer - 1 i minne 69, forat eksekvering av (A) skal skje i minne 00 første gang. Tømmer minnene 67 og 68. (Telling av hvor mange enheter som oppfyller kravene, evt. sum x.) Ved RST-funksjonen fjernes alle flagg - derfor er en R/S instruksjon også lagt i begynnelsen av programmet, ved steg 000.

Lbl E' 260  
Foretar ny oppdeling - til 339,69. Alle minner tømtes, og det gis utskrift: "VARIABLE-DIGIT EQ" for identifikasjon av programmet. "FLAG" lagres i Op 04, for bruk i flaggsettingsprosedyrene.

Lbl A' B' C' D' og D:  
(220 - 228 - 236 - 244 - 252)  
Setter flagg for identifikasjon av hvilken testfunksjon som velges, og for automatisk utskrift av styringsminner som skal vere tomme i to tilfeller.  
A': Flagg 0. Gir utskrift når to spesifiserte betingelser er oppfylt.  
B': Flagg 1. Gir utskrift når en av to spesifiserte betingelser er oppfylt.  
C': Flagg 2. Gir utskrift når en første spesifikasjon ikke oppfylles, men en andre oppfylles.  
D': Flagg 3. Gir utskrift når en spesifisert betingelse oppfylles.  
D: Flagg 4. Gir utskrift for alle dataenheter av en variabel, eller flere sammenhengende. I utskrift finnes også  $\sum x$  og  $\bar{x}$  for variabelen.

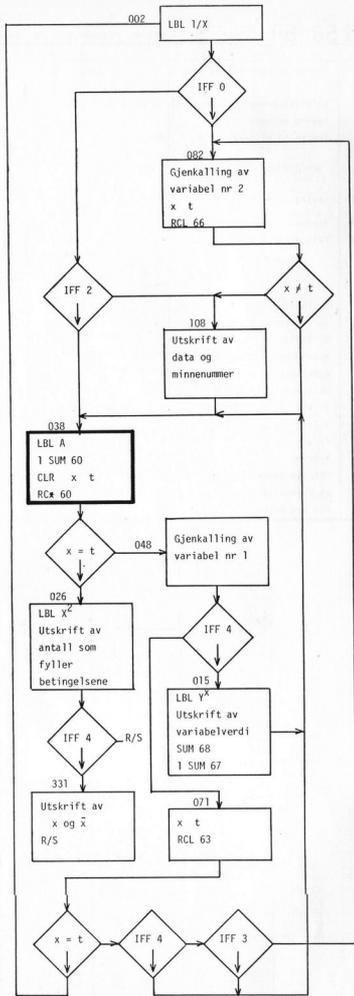


DIAGRAM 2

MINNE-INNHOLD: (konstruert eksempel, med maskinen i ulik posisjon ved beregning)

1230000000.	00	}	DATA
2410000000.	01		
1235000000.	02		
1230000000.	03		
2510000000.	04		
1254000000.	55	}	DATA
1254000000.	56		
2143000000.	57		
2143000000.	58		
1210000000.	59		
0.	60		
10000000.	61		
10.	62		
1.	63		
10000000.	64		
10.	65		
1.	66		
0.	67		
5.	68		
	69		

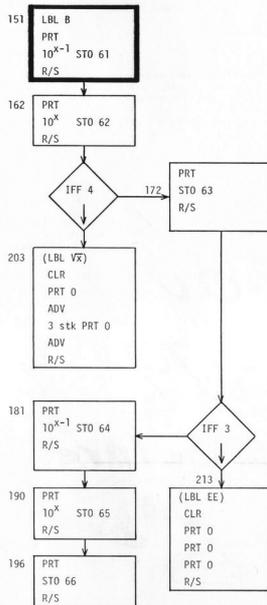


DIAGRAM 1

← 1 → 2 → INSTRUMENTS → 7						
VARIABLE - DIGIT EQUATION						
-and-	-or-	n-but	SING.	INIT.		
RUN	VAR	EQ	DATA	PRT.	RESET	

LABELS

002	35	1/X
015	45	V <sup>x</sup>
026	33	x <sup>2</sup>
038	11	A
121	13	C
137	15	E
151	12	B
202	34	F <sup>x</sup>
212	32	E
220	16	A'
228	17	B'
236	18	C'
244	19	D'
252	14	D
260	10	E'

ANDRE BRUKSMULIGHETER

En mer hobbypreget mulighet, er å bruke programmet til enkle kartotek- og katalogiseringsformål - f.eks. en platesamling. Artist, instrument mv. gis koder, og platene nummereres fortløpende. Om nyttevarden er begrenset, er et slikt eksperiment lærerikt.

NYE PROGRAMMER

Det vil bli utarbeidet programmer med samme innlesingsprosedyre, oppdeling mv., men med alternative behandlingsmåter, særlig med sikte på å føre data direkte over til statistisk behandling i funksjonene i standardmodulen eller statistikkmodulen.



ROULETT

A	B	C	D	E	F	G
Steg	Instruktion	Exempel	Mata in	Tryck ned	Display	Printer
<u>Initiering</u>						
1	Uppdelning av minnesrymme		5	2nd op	17 559.49	
2	Läs alla fyra kortsidorna					
<u>Spelstart</u>						
3 a	Om alla spelare ska ha samma startkapital		Startkapital	2nd E'	Startkapital	
		Alla spelare ska ha 50:- som startkap	50	2nd E'	50.	
3 b	Om spelarna ska ha olika startkapital		Startkap för spelare A " " " B " " " C " " " D	Sto 10 " 11 " 12 " 13		
		Spelare A ska ha 200:- " B " " 100:- " C " " 50:- " D " " 150:-	200 100 50 150	Sto 10 " 11 " 12 " 13	200. 100. 50. 150.	
<u>Satsning</u>						
4	Satsningar för spelare A		Insats.kod	A	Kapital för spelare A efter insats	
		Spelare A har 200:- och vill satsa 50:- på mellersta duss.	50.45	A	150.	

A	B	C	D	E	F	G
5	Om spelare A vill satsa på fler saker. Detta kan upprepas högst 5 ggr		Insats.kod	R/S	Kapital för spelare A efter insats	
		Och 10:- på nummer 35-36	10.85	R/S	140.	
		Och 50:- på 3:e kolumn	50.49	R/S	90.	
		Och 50:- på de 18 sista numren	50.43	R/S	40.	
		Och 20:- på udda numren	20.41	R/S	20.	
		Och 10:- på nummer 20	10.20	R/S	10.	
6	Satsningar för spelare B		Insats.kod	B	Kapital för spelare B efter insats	
		Spelare B har 25:- och vill satsa 5:- på nr 35	5.35	B	20.	
7	Om spelare B vill satsa på fler saker		Insats.kod	R/S	Kapital för spelare B efter insats	
8	Satsningar för spelare C		Insats.kod	C	Kapital för spelare C efter insats	
		Spelare C har 50:- och vill satsa 10:- på 3:e kolumnen	10.49	C	40.	

A	B	C	D	E	F	G
9	Om spelare C vill satsa på fler saker		Insatskod	R/S	Kapital för spelare C efter insats	
10	Satsningar för spelare D		Insatskod	D	Kapital för spelare D efter insats	
		Spelare D har 200:- och vill satsa 7:- på nr 0		7.37	193.	
11	Om spelare D vill satsa på fler saker		Insatskod	R/S	Kapital för spelare D efter insats	
<u>Anmärkning</u>						
		Till steg 4-11: Varje spelare kan satsa på 70lka saker. Steg 5,7,9,11 kan alltså upprepas 5 ggr för varje spelare. Om man är ensam hoppar man över steg 6-11. Om det är 2 spelare hoppar man över steg 8-11. Om det är 3 spelare hoppar man över steg 10-11.				

A	B	C	D	E	F	G
12	När alla satsat klart					Siffran rouletten stannar på, sedan ROUGE/NOIR och i vilken kolumn numret finns (1-3).
						Numret som rouletten stannar på, sedan ROUGE/NOIR och i vilken kolumn numret finns (1-3).

A	B	C	D	E	F	G
13	Kalkylatorn kollar automatiskt alla insatser och summerar eventuella vinster		Rouletten stannar på 20, som är svart och finns i andra kolumnen  Rouletten stannar på 34 som är rött och finns i första kolumnen		20. 2.  34.00 1.	20 NOIR 34 ROUGE 1
			Spelare B vinner 40:-		Om någon vunnit visas först ett nummer (1-4) på vem som har vunnit. 1=spelare A, 2=spelare B osv...	Om skrivare är ansluten så skrivs vem som vinner vad.
14	För att se hur mycket spelare A har kvar			2nd A'	2.	40 B
15	För att se hur mycket spelare B har kvar			2nd B'		Hur mycket spelare A har kvar
16	För att se hur mycket spelare C har kvar			2nd C'		Hur mycket spelare B har kvar
17	För att se hur mycket spelare D har kvar			2nd D'		Hur mycket spelare C har kvar
18	För att spela en gång till repetera från och med steg 5					Hur mycket spelare D har kvar
19	Om ni istället vill fortsätta en annan gång med samma kapital		Kortsida 4	4 2nd WRITE		
<u>Anmärkning</u>						
		Steg 14-17 kan utföras i valfri ordning. Detta gäller också steg 5,7,9 och 11.				

PROGRAMLISTNING

ROULETT

000	01	GT	0
001	62	GD	2
002	26	26	6
003	76	LBL	0
004	48	EXC	0
005	02	2	2
006	49	PRD	0
007	14	14	14
008	76	LBL	0
009	42	STD	0
010	19	19	19
011	02	02	2
012	85	+	+
013	03	3	3
014	73	-	-
015	69	DP	0
016	04	04	4
017	04	1	1
018	02	2	2
019	95	=	=
020	66	PRU	0
021	66	PRU	0
022	13	RCL	0
023	14	14	14
024	69	DP	0
025	06	06	6
026	66	PRU	0
027	66	PRU	0
028	74	SM*	0
029	02	02	2
030	76	LBL	0
031	45	YX	0
032	69	DP	0
033	61	21	21
034	97	DSZ	0
035	03	03	3
036	33	X*	0
037	76	LBL	0
038	39	CDS	0
039	43	RCL	0
040	03	03	3
041	44	SUM	0

MINNES LISTNING

5.	00
8.	01
12.	02
13.	03
3132243500.	04
38324121217.	05
12.	06
13.	07
44.	08
4287.16302.	09
200.	10
300.	11
200.	12
105	13
106	14
105	15
0.	16
0.	17
0.	18
0.	19
0.	20
0.	21
0.	22
0.	23
0.	24
0.	25
0.	26
0.	27
0.	28
0.	29
0.	30
0.	31
0.	32
0.	33
0.	34
0.	35
0.	36
0.	37
0.	38
0.	39
0.	40
0.	41
0.	42
0.	43
0.	44
0.	45
0.	46
0.	47
0.	48
0.	49
0.	50

120	83	GD	208	06	6	296	01	01	384	43	RCL	472	01	1
121	15	15	209	08	8	297	61	GT	385	00	00	473	03	3
122	04	4	210	42	ST	298	23	LNX	386	95	=	474	42	ST
123	09	9	211	08	08	299	43	RCL	387	95	=	475	15	15
124	05	INW	212	61	GT	300	05	388	02	2	2	476	08	08
125	77	GE	213	71	SBR	301	69	DP	389	95	=	477	42	ST
126	01	01	214	03	3	302	01	01	390	32	XIT	478	06	06
127	85	85	216	02	ST	303	57	ENG	391	43	RCL	479	03	03
128	06	6	217	05	5	304	02	02	392	00	00	480	14	14
129	06	6	217	05	5	305	02	02	393	55	+	481	76	LBL
130	22	INW	218	06	6	306	61	GT	394	02	2	482	25	CL
131	77	GE	219	4	4	307	03	03	395	59	INT	483	14	14
132	01	01	220	08	08	308	13	08	396	59	INT	484	14	14
133	71	71	221	61	GT	309	43	RCL	397	67	EXC	485	38	SIN
134	01	1	222	71	ST	310	04	04	398	48	EXC	486	50	INT
135	03	3	223	67	GT	311	69	DP	399	61	GT	487	44	SUM
136	01	01	224	15	E	312	01	01	400	45	YX	488	09	09
137	06	6	225	81	RST	313	43	RCL	401	22	INW	489	22	INW
138	04	4	226	36	PGH	314	00	00	402	87	DSZ	490	97	DSZ
139	04	4	227	1	1	315	66	PRU	403	03	03	491	06	06
140	22	INW	228	71	SBR	316	66	PRU	404	45	YX	492	66	PRU
141	08	08	229	88	DMS	317	66	PRU	405	61	GT	493	43	RCL
142	76	LBL	230	65	X	318	66	PRU	406	48	EXC	494	14	14
143	71	SBR	231	03	3	319	66	PRU	407	76	LBL	495	59	INT
144	43	RCL	232	07	7	320	22	INW	408	11	R	496	32	XIT
145	00	00	233	95	9	321	57	ENG	409	42	ST	497	73	RC*
146	09	9	234	19	INT	322	42	ST	410	14	14	498	15	15
147	01	1	235	98	ADV	323	53	+	411	01	1	499	22	INW
148	95	=	236	99	PRT	324	53	+	412	06	6	500	77	GE
149	54	54	237	42	ST	325	24	CE	413	42	ST	501	66	PRU
150	43	RCL	238	00	0	326	08	08	414	08	08	502	43	RCL
151	06	06	239	69	DP	327	93	+	415	01	1	503	14	14
152	95	=	240	00	00	328	01	1	416	00	0	504	72	ST*
153	95	=	241	00	00	329	94	+	417	42	ST	505	08	08
154	84	84	242	01	1	330	01	1	418	15	15	506	69	DP
155	43	RCL	243	08	8	331	03	3	419	08	8	507	28	28
156	08	08	244	02	ST	332	54	+	420	42	ST	508	59	INT
157	43	RCL	245	03	3	333	59	INT	421	06	6	509	06	06
158	22	INW	246	00	0	334	45	+	422	43	RCL	510	74	SM*
159	67	EQ	247	42	ST	335	03	3	423	14	14	511	15	15
160	45	YX	248	02	ST	336	85	+	424	13	13	512	15	15
161	43	RCL	249	4	4	337	42	ST	425	25	CL	513	15	15
162	06	6	250	42	ST	338	07	7	426	76	LBL	514	92	RTN
163	55	-	251	01	01	339	66	PRU	427	18	B	515	61	GT
164	43	RCL	252	76	LBL	340	66	PRU	428	12	14	516	25	CL
165	06	06	253	03	3	341	05	5	429	14	14	517	76	LBL
166	95	=	254	02	2	342	95	=	430	02	2	518	10	E*
167	49	PRD	255	65	X	343	69	DP	431	03	3	519	42	ST
168	43	RCL	256	43	RCL	344	03	3	432	42	ST	520	10	10
169	03	3	257	03	03	345	03	03	433	08	08	521	42	ST
170	42	ST	258	75	-	346	67	EQ	434	01	1	522	11	11
171	03	3	259	43	RCL	347	03	03	435	01	1	523	42	ST
172	PRD	3	260	02	2	348	70	70	436	02	2	524	12	12
173	14	14	261	95	=	349	69	DP	437	15	15	525	42	ST
174	43	RCL	262	67	EQ	350	05	05	438	08	8	526	13	13
175	07	7	263	02	2	351	01	1	439	42	ST	527	25	CL
176	XIT	9	264	09	9	352	09	9	440	06	6	528	15	15
177	75	75	265	22	INW	353	07	GE	441	43	RCL	529	76	LBL
178	04	4	266	97	DSZ	354	03	03	442	14	14	530	19	D*
179	06	6	267	03	03	355	58	58	443	61	GT	531	01	1
180	22	INW	268	03	3	356	03	3	444	44	44	532	85	+
181	67	EQ	269	09	09	357	03	03	445	76	LBL	533	76	LBL
182	42	ST	270	97	DSZ	358	01	1	446	13	C	534	18	C*
183	21	21	271	01	01	359	06	6	447	42	ST	535	01	1
184	45	YX	272	23	LNX	360	42	ST	448	14	14	536	85	+
185	05	5	273	87	IFF	361	03	03	449	03	3	537	76	LBL
186	5	5	274	01	01	362	25	CLR	450	00	0	538	17	B*
187	28	LDC	275	28	LDC	363	42	ST	451	02	2	539	01	1
188	7	7	276	05	5	364	01	1	452	08	08	540	85	+
189	02	02	277	42	ST	365	09	9	453	01	1	541	76	LBL
190	00	00	278	01	01	366	42	ST	454	02	2	542	16	R*
191	61	GT	279	1	1	367	02	2	455	79	ST	543	02	2
192	28	LDC	280	42	ST	368	61	GT	456	15	15	544	00	0
193	06	06	281	02	2	369	39	CDS	457	08	8	545	85	+
194	05	5	282	86	STF	370	86	STF	458	05	5	546	07	GT
195	01	01	283	01	01	371	71	71	459	06	6	547	07	GT
196	42	ST	284	61	GT	372	61	GT	460	43	RCL	548	07	7
197	08	08	285	23	LNX	373	03	03	461	14	14	549	07	7
198	61	GT	286	76	LBL	374	58	58	462	61	GT	550	02	2
199	01	01	287	28	LDC	375	22	INW	463	25	CLR	551	03	3
200	6	6	288	04	4	376	87	IFF	464	76	LBL	552	95	=
201	7	7	289	42	ST	377	02	2	465	14	14	553	69	DP
202	02	02	290	01	01	378	02	2	466	42	ST	554	02	2
203	02	02	291	00	0	379	61	GT	467	14	14	555	73	RC*
204	14	14	292	42	ST	380	48	EXC	468	03	3	556	07	7
205	02	02	293	02	2	381	01	1	469	07	7	557	69	DP
206	42	ST	294	42	ST	382	24	24	470	42	ST	558	06	06
207	06	06	295	86	STF	383	85	+	471	08	08	559	91	R/S

# PROGRAMBIBLIOTEKET

Varje program i programbiblioteket beskrivs kortfattat enligt följande mall:

( Beskrivande namn, max 80 nedslag ) ( ) ( ) ( ) ( )  
 ( Kortfattad beskrivning av vad programmet utträttar samt huvudsakliga förtjänster med programmet, max 500 nedslag )  
 (( Utan skrivare )) ( Interaktivt=502 ) ( Tot.Ex.tid=1234s ) ( Med.vintetid=123s ) ( Modu=2 ) ( Sidor=-1,-2,-3, 4 ) ( Steg=687 ) ( Reg.=23 )

( programkod 4 pos. ) ( maskin 4 pos. ) ( ågare 4 pos. ) ( pris 6 pos. )

Följande varianter finns:  
 skrivare nödvändig = Med skrivare  
 kan köras utan skrivare = (Med skrivare)  
 viss utskrift med skrivare = (Utän skrivare)  
 ingen utskrift = Utän skrivare

Talar om hur stor del av exekveringsinstruktionerna som skrivs ut under körning  
 Total exekveringstid  
 Medelväntetid mellan åtgärder  
 Talar om i fall någon biblioteksmodul måste vara isatt  
 Antal inspelade kortsidor (- innebär spårade kortsidor)  
 Antal programsteg  
 Antal använda register

## VÄGLEDNING VID MEDLEMSBESTÄLLNING

OBSERVERA ATT TILL PRISET I PROGRAMBIBLIOTEKET KOMMER EN FÖRMEDLINGSAVGIFT PÅ 60:- PER UPPDRAG!

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 uthämtat mot postförskott. Kontant betalning kan ej accepteras.

De priser som anges i programbiblioteket är vad du betalar för att få programmet ner i brevlådan.

Kom ihåg att:

- betalning sker till postgiro 430 01 59 - 3 och lämpningen på i denna tidning instoppade balketter.
- ange önskad vara med den referenskod som är knuten till varan i tidningen. I programbiblioteket står

dena kod på samma rad som programmet.

- leverans sker till den adress som finns angiven på postgiroinbetalningskortet eller, om endast medlemskod är angiven, till denna adress.

Vid leverans av program ur programbiblioteket får du:

- en kopia av ifyllt mall för inskickade program i förekommande fall.
- programbeskrivning enligt mallens punkt 4: komplett för ej spårade program, för spårade program utgått 4e - 4h
- hela programmet på programmerade magnetkort.

Observera som betalar på följande mall:		PROGRAMMkod	ÅR	PRIS
Spelplan till 3D Luffarschack		85ac	TI-59	40:-
PLAN 1				
*10*11*12*13+ *14*15*16*17* *18*19*20*21* *22*23*24*25*		Programmet skriver ut på skrivaren en spelplan till det 3D-Luffarschack som finns beskrivet i Programbiten 78-2.		
Interakt.	tot.exek.	med.vänt.	Modul	Antal
2	1.18	min	ingen	1,2,3
762	762	762	762	762
Programmerare	År	Pris	År	Pris
Q1m6	Q1m6	Q1m6	Q1m6	Q1m6
2. Beskrivande namn kortare än 80 tecken:				
Transientförlopp i R,L,C - länk		PROGRAMMkod	ÅR	PRIS
85ad		58/59	40	
		Programmet beräknar stegvisare för en krets enligt figuren. Resultatet plottas ut på PG-100 skrivare. Den ekvation som löses är: $v'' + 2\zeta\omega_0 v' + \omega_0^2 v = u_0^2 u$ där $\zeta = \frac{R}{2L}$ , $\omega_0 = \frac{1}{\sqrt{LC}}$		
Interakt.	tot.exek.	med.vänt.	Modul	Antal
2	3,5	min	ingen	2
275	275	275	275	275
Programmerare	År	Pris	År	Pris
Q1o5	Q1o5	Q1o5	Q1o5	Q1o5
3. Beskrivande namn kortare än 80 tecken:				
(-) 0 (+) Biorytmkurvor		PROGRAMMkod	ÅR	PRIS
85ab		TI-59	100	
		Programmet plottar alla tre biorytmkurvorna samtidigt tillsammans med viss ledtext.		
Interakt.	tot.exek.	med.vänt.	Modul	Antal
2	0,2	min	30	2
451	451	451	451	451
Programmerare	År	Pris	År	Pris
P21b	P21b	P21b	P21b	P21b
4. Beskrivande namn kortare än 80 tecken:				
Nyktarhetsprogram		PROGRAMMkod	ÅR	PRIS
85ag		TI-59	40:-	
Beräknar med utgångspunkt från intagen mängd sprit, vin och öl aktuell alkoholhalt i blodet samt beräknad tid till helt nykter. Tar hänsyn till kön och kroppsvolym. Helt självinstruerande text.				
Interakt.	tot.exek.	med.vänt.	Modul	Antal
2	0,2	min	30	2
451	451	451	451	451
Programmerare	År	Pris	År	Pris
P21b	P21b	P21b	P21b	P21b
5. Beskrivande namn kortare än 80 tecken:				
Allmänna gaslagen (Gasernas allmänna tillståndslag)		PROGRAMMkod	ÅR	PRIS
85ae		TI-58/59	40:-	
Matas in tre av variablerna p=tryck(pascal),V=volym(m <sup>3</sup> ),n=mubotantmängden(antalet mol), T=absolut temperatur(kelvin), så beräknar räkneren den variabeln som inte matats in. Programmet räknar med SI enheter.				
Interakt.	tot.exek.	med.vänt.	Modul	Antal
2	0,2	min	30	2
451	451	451	451	451
Programmerare	År	Pris	År	Pris
P21b	P21b	P21b	P21b	P21b
6. Beskrivande namn kortare än 80 tecken:				
Regressions-analys		PROGRAMMkod	ÅR	PRIS
85aa		TI-59	100	
Regression är ett effektivt sätt att finna ett samband mellan en oberoende variabel (x) och en beroende variabel (y). Detta program utför inte mindre än fem olika regressioner (linjär, exponentiell, rot, kvadrat, invertering) samtidigt, och väljer sedan ut den bästa. Du kan även själv välja vilken typ av regression du vill ha.				
Interakt.	tot.exek.	med.vänt.	Modul	Antal
2	0,4	15	min	1
479	479	479	479	479
Programmerare	År	Pris	År	Pris
P21b	P21b	P21b	P21b	P21b
7. Beskrivande namn kortare än 80 tecken:				
DATA = LAGRER = GJENKALLING OCH INNEHÖLSTESTING		PROGRAMMkod	ÅR	PRIS
85af		59	40:-	
Lagrer og jenkaller data - sexti enheter på magnetkort. Hver enhet kan ha ti variabler + identifikationsiffre; hver variabel 10 verdier. I eller 2 variabler med ett eller flere sifre og tilhørende betingelser testes for:(a og b); (a eller b); (ikke a men b). En variabel testes alene, eller utskrift av en variabel med beregning av Ix og X. Benyttes til bearbejdning av data fra spørreskjema og liknande eller til enkle katalogiseringsformål				
Interakt.	tot.exek.	med.vänt.	Modul	Antal
2	0,4	15	min	1
479	479	479	479	479
Programmerare	År	Pris	År	Pris
P21b	P21b	P21b	P21b	P21b

# REKOMMENDERADE återförsäljare:

Under den här rubriken låter vi de återförsäljare som är medlemmar i föreningen annonsera med Firmanamn, adress, postadress och telefonnummer.

KONTORSSYSTEM AB, Skeppsbrohuset  
411 18 GÖTEBORG. Tel.: 031-174455

GUMPERTS AB, Polstjärnegatan 4  
402 25 GÖTEBORG. Tel.: 031-235480

JANKEN MINIDATA, Gibraltargatan 22  
411 32 GÖTEBORG. Tel.: 031-180250

WETTERGRENS BOKHANDEL, V. Hamngatan 22  
Avenyn 21, Vasagatan 22, Frölunda torg  
Wieselgrensplatsen, GÖTEBORG  
Kungsmässan, KUNGSBACKA. Tel.vx: 031-170090

DILLBERGSKA BOKHANDELN, Kaggensgatan 16  
392 32 KALMAR. Tel.: 0480-10164, 11011

BJÖRKMANS AB, Stensättravägen 5-7  
127 03 SKÄRHOLMEN. Tel.: 08-970540  
Humlegårdsgatan 22, STOCKHOLM. Tel.: 626581

ELIKON, Regeringsgatan 30  
111 53 STOCKHOLM. Tel.: 08-219300

LINDAHL & ROTHOFF AB, Östra Kristinelundsvägen 16  
200 71 MALMÖ. Tel.: 040-101730

MASKINAFFÄREN FYRIS: Kungsgatan 32  
753 21 UPPSALA. Tel: 018-149015