

PROGRAM

nittinian

BITEN

FORTH
spalten

REDAKTÖR: Lars-Erik Svahn

85-1

STADGAR

LT-WRITER

Trippelpunkten

PASCAL

Innehåll

innehåll pb 85-1

Ordföranden - redaktören	2
Stadgar för Programbiten	3
Årsmötesprotokoll	4
Styrelsens årsredovisning	4-5
Resultat och balans	5
Utmaningen	6-10
Oj då!, Oj-oj då! Oj-oj-oj då!	10-11
Material till tidningen	11
II-59 Koefficienter för andra- gradsekvation	12-13
Vilket kägelsnitt anger andragradsekvationen	14-15
FORTH-spalten	16-17
Catalog	17
Trippelpunkten på 99:an	18-20
Nya program och programversioner	20-21
LT-writer	22-25
Konstiga tecken	26
Upprop PASCAL !	26
Programbanken	28-27

ISSN 0281-1146

ORDFÖRANDEN Redaktören

Och den tillfällige har ORDET

Den 23 februari hade föreningen årsmöte. I tidningen redovisar vi nu styrelsens verksamhetsberättelse för 1984 med tillhörande ekonomisk redovisning, årsmötesprotokollet och förslaget till nya stadgar som behandlades på mötet.

Som framgår har vi fått en ny styrelse. Jag har fått förtroendet att leda styrelsen ett år till, vilket jag tackar för. Jag vill också taacka de avgående styrelseledamöterna och vänder mig särskilt till Bengt Fahlgren och Björn Mårtensson, som jobbat bra som kassör respektive programbankir. Björn, som lämnar oss för att bli professionell i databranschen på helt andra maskiner, önskar vi lycka till. Bengt önskar vi se artiklar och annat av även i fortsättningen, och det har han sagt att han strävar att åstadkomma.

Till de nya styrelseledamöterna säger jag välkommen:
- Gösta Blume, kassör och sina 76 år till trots aktiv med att lära sig 99-an. Han har ett förflutet som aktiv med programmerbara räknare

- Jan Alexandersson, som vi sett spår av i spalterna som tipsare om tricks, som programmerare med ett starkt utforskande intresse och som författare

- Börje Häll, som vi sett i aktion som praktiskt inriktad, skicklig programmerare.

Göran Nygren lämnar redaktörssysslan av tidsskäl men tar över programbanken efter Björn Mårtensson. Han tänker bl a se till att alla program som publiceras i tidningen blir tillgängliga i banken. Vi har haft dåliga rutiner för det tidigare. Han avser också verka som distributör av de nya versioner mm av program från Texas vi kommer att tillhandahålla.

Vi andra håller på som förut, är det tänkt. Dock behöver vi lösa frågan om redaktör för tidningen.

Valberedning och årsmöte misslyckades med det. För det här numret har jag tagit hand om saken, men det är inte hållbart i längden. Vi resonerar med en del personer men behöver fler uppslag och kontakter. Tag Ditt ansvar, käre medlem, och bidrag på något sätt i den frågan. Ett bidrag får vi från Lars-Erik Svahn, som med början i det här numret står för en spalt om FORTH, som han av premiärspalten att döma älskar över alla andra programmeringsspråk. Anders Persson fortsätter med Utmaningen, för vilket alla medlemmar har anledning vara mycket glada, och Claes Schibler håller i 59-sidorna förutom allt annat han gör. Jag misstänker att FORTH-spalten och Utmaningen kan kollidera om ämnen och åsikter, men jag tycker det är bara bra med debatt och meningsbrytningar. Och det gäller förstås inte bara herrar Persson och Svahn utan Dig också!

När väl Göran Nygren kommit igång som programbankir räknar jag med att han skall presentera program i tidningen och berätta vad som händer i banken. Flera personer har lovat att skriva särskilt för dem som har basmaskinen enbart och behandla kassetthantering och Basic/X-Basic. Det syns bra i det här numret med Lennart Thelander's ordbehandlare LT-Writer, t ex.

Lägg märke till de nya program vi fått från TI i USA genom Maurice E. T. Swinnen. Anders Persson presenterar dem i en artikel. Vi trycker också av den bruksanvisning som följer med när Du köper kopiorna av programmen. Du kan beställa dem från föreningen som vanligt.

Vi har talat en del om Särskilda IntresseGrupper (SIG). Som jag uppfattar det finns det intresse bl a av

- adventure-spel, där Johan Bleeker i Lidköping och Love Feuer är mest aktiva

- FORTH, där Lars-Erik Svahn kan hålla samman verksamheten

- assembler och hårdvara, där Michael Öhman och Tony Rogvall ser ut att vara mest aktiva

- PASCAL, där Anders Persson har ett upprop i denna tidning

- telekommunikation och databaser, där Michael Öhman och Joakim Söderberg är entusiasterna

Jag hoppas på aktivitet i SIG-arna. Tag initiativ om Du är intresserad!

Nu har vi äntligen fått en föreningens telefon. Den passas t v på tisdagar klockan 19-21 på kvällen. T v passar Michael Öhman. Telefonnumret är 08/761 51 62.

Tala med Michael om Du har ett föreningsärende!

Vi bör rekrytera fler medlemmar. Ju fler vi är desto fler ideer finns det att dela på. Det är planerat en artikel i Mikrodatorn nr 3 1985 om föreningen, bl a. Jag har nu fått litet bättre kontakt med Volvo-folket också. Det resoneras om ett medlemsmöte i Göteborg. Har Du ideer eller kan Du bidra på något sätt? Hör av Dig till mig eller till Douglas Lundqvist i Göteborg. Han kan nog ordna med en bra lokal, t ex.

Happy computing!

Lennart Lindberg
Lennart Lindberg, ordförande och t redaktör

PS Har jag talat om att Björn Gustavsson nu ställt källkoden för de utökningar till TI 2.0, som ger föreningens FORTH, till förfogande och att vi kommer att distribuera den på samma sätt som annan FORTH? Det innebär att Du beställer på vanligt sätt. Just när jag skriver det här är det oklart med priset. Det rör sig om två disketter. Hör av Dig om Du är intresserad! DS

PPS NÄSTA NUMMER: Björn Gustavsson berättar om sin FORTH-kod. Hans Wickström talar om hur man bygger ett RS232 billigt för joystick-porten. Jan Alexandersson har hittat intressanta saker om modulen Personal Record Keeping och de filer den arbetar med. Anders Persson beskriver de kluriga assembler-rutiner han skrivit till LT-Writer. mm! DS

I REDAKTIONEN

Redaktör	Lennart Lindberg
Utmaningsredaktör	Anders Persson
Forth-redaktör	Lars-Erik Svahn
Programförmiddlare	Göran Nygren
Allt-i-allo	Claes S

Detta nummer av PB hand-assemblerat av Claes

Föreningens och redaktionens adress:

Föreningen Programbiten	+++++
c/o Schibler	+ DATAINSPEKTIONENS +
Wahlbergsgatan 6 1 tr ned	+ LICENSNUMMER +
121 46 Johanneshov	+ +
	+ 82100488 +
Postgiro 19 83 00 - 6	+++++

Medlemsavgift för 1985 är120 SEK
Nittinian, årgång 1983 (nr 1, 2, 3, 4/5) kostar 80 SEK
Programbiten - Nittinian 1984, 1 - 5 -"- 100 SEK

ANVÄNDARTIPS MED MINI MEMORY

Denna bok skrevs av Björn Gustavsson på uppdrag av Texas Instruments. Precis när boken var klar lade TI ned hemdatorn. Föreningen fick då rätten till boken och har tryckt upp den. Boken innehåller 60 sidor och kostar för medlemmar 60 SEK som sätts in på föreningens postgiro 19 83 00 - 6.

Annonser, insatta av enskild medlem (ej företag), som gäller försäljning av moduler eller andra tillbehör i enstaka exemplar är gratis.

Övriga annonser kostar 2000 SEK per helsida, förutom baksidan som kostar 3000 SEK.

För kommersiellt bruk gäller följande:

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

STADGAR

STADGAR FÖR FÖRENINGEN PROGRAMBITEN

Årsmötets beslut den 23 februari 1985.

Kapitel 1. Föreningens syfte.

Föreningen Programbiten har till syfte att samla personer med intresse för programmerbara räknare (företrädesvis TI-59) och person- och hemdatorer (företrädesvis TI-99/4A) i sådana former att de kan hjälpa varandra med avseende på maskiner, tillbehör, program, användningsområden mm.

Föreningen utger, för att främja detta syfte, medlemstidningen PROGRAMBITEN, som i huvudsak avses behandla nämnda datorer och räknare.

Föreningen skall i grunden vara icke kommersiell. Detta skall inte hindra föreningen att göra marginella affärer som gynnar medlemmarna.

Kapitel 2. Föreningens sammansättning.

Som medlem räknas den som betalt av årsmötet fastlagd medlemsavgift.

Medlem, som bryter mot föreningens stadgar eller på annat sätt uppenbarligen skadar föreningen eller dess syften, kan av styrelsen med enhälligt beslut uteslutas ur föreningen.

Kapitel 3. Föreningens verksamhetsår.

Programbitens verksamhetsår och funktionärernas mandattid är kalenderåret (1 januari - 31 december).

Kapitel 4. Föreningens exekutiva organ.

Programbitens exekutiva organ är:

- Föreningsstämman, - ordinarie (= årsmöte)
- Föreningsstämman, - extra
- Styrelsen
- Revisorerna
- Årsmötets valberedning.

Kapitel 5. Programbitens funktionärer.

Årsmötet väljer följande funktionärer:

- Styrelse, bestående av ordförande, sekreterare, kassör och redaktör, vilka samtliga utses till befattning av årsmötet, samt därutöver minst en och högst fem ledamöter i övrigt.
- Två revisorer och en revisorsuppleant.
- Valberedning bestående av två ledamöter, varav en sammankallande.

Övriga funktionärer utses av styrelsen.

Programbitens funktionärer skall vara medlemmar i föreningen. Undantagna är revisorerna, som kan vara utomstående.

Funktionärerna är ansvariga inför det organ som utsett dem.

Kapitel 6. Stämman.

Stämman är föreningens högsta beslutande myndighet och bildas av föreningens samtliga medlemmar, som alla har en röst vardera. Stämmans ordförande har utslagsröst utom i valfrågor. Minst en stämma skall hållas årligen.

Årsmötet, dvs ordinarie stämma, skall hållas senast 15 mars.

Stämma är beslutsmässig med det antal medlemmar som infinner sig, förutsatt att kallelse med tid och plats skriftligt utsänts till medlemmarna antingen genom tidningen eller separat, senast 15 dagar innan sammanträdet.

Årsmöte sammankallas av styrelsen.

Extra stämma sammankallas av styrelsen eller av revisorerna på eget initiativ eller när minst 30 medlemmar så fordrar. Ingen får delta i beslut eller leda sammanträde när fråga om ansvarsfrihet för honom/henne själv behandlas.

Vid årsmöte förekommer ärenden, som av styrelsen hänskjutits till årsmötet eller som skriftligen inlämnats till styrelsen av medlem senast sista dagen av det verksamhetsår som skall redovisas på mötet. Beslut kan endast fattas i frågor som upptagits på föredragningslistan.

Årsmötet skall behandla följande frågor:

- Styrelsens berättelse över föregående verksamhetsår.
- Revisorernas berättelse över föregående verksamhetsår.
- Ansvarsfrihet för styrelsen och funktionärerna.
- Val av styrelse med funktionärer.
- Val av revisorer.
- Val av representanter i valberedningen.

Protokoll skall föras som beslutsprotokoll och skall justeras inom 20 dagar efter mötet av sammanträdesledaren jämte två av årsmötet utsedda justeringsmän. Protokollet skall i justerat skick utsändas till samtliga medlemmar.

Kapitel 7. Föreningsstyrelsen (FS).

FS består av föreningens ordförande, kassör, sekreterare och redaktör samt minst en ledamot därutöver.

FS är beslutsmässig, om minst tre ledamöter är närvarande och om kallelse med tid och plats meddelats senast tre arbetsdagar innan mötet.

FS ordförande har utslagsröst vid lika röstsliffor.

Protokoll skall föras. Protokollet justeras av ordförande och utskickas till styrelsemedlemmarna.

Det åligger FS:

att inför årsmötet ansvara för hela föreningens förvaltning och verksamhet.

att verkställa årsmötets beslut.

att handha och ansvara för föreningens medel, bereda inkomna förslag, handha föreningens korrespondens samt i övrigt sköta löpande ärenden.

att vid behov förrätta interimisval.

att före utgången av varje verksamhetsår tillse att gällande stadga är utskriven.

att senast 20 dagar före ordinarie stämma överlämna föreningens årsredovisning till revisorerna.

Kapitel 8. Föreningens revisorer.

Revisorerna

skall granska föreningens förvaltning och verksamhet.

skall senast fem dagar före möte om ansvarsfrihet till FS inlämna revisionsberättelse.

Revisorerna äger rätt

att när helst de så önskar taga del av samtliga räkenskaper, protokoll och andra handlingar.

att begära och erhålla upplysningar rörande verksamhet och förvaltning.

att övervara samtliga föreningsorgans sammanträden med yttranderätt.

att inkalla samtliga föreningsorgan.

Kapitel 9. Föreningens valberedning.

Årsmötets val förbereds av valberedningen.

Valberedningens förslag skall inlämnas till FS senast tolv dagar innan årsmöte. Förslaget bör helst distribueras med kallelsen till årsmötet.

Kapitel 10. Ändring av stadgarna.

Förslag till ändring av dessa stadgar inlämnas skriftligen till FS senast tolv arbetsdagar före årsmötet, vid vilket FS utlåtande skall föreligga. För slutgiltigt bifall fordras beslut på två på varandra följande stämmor, varav minst ett årsmöte. På det möte som sist hålles skall beslutet ha bifallits med minst 2/3 majoritet.

Kapitel 11. Tolkning av stadgarna.

Årsmötet äger rätt att tolka dessa stadgar. FS kan i samråd med revisorerna avgöra i årsmötets ställe.

Kapitel 12. Föreningens upplösning.

Beslut om föreningens upplösning kan endast fattas på stämma. För slutgiltigt bifall fordras beslut på två på varandra följande stämmor, varav minst ett årsmöte. På det möte som sist hålles skall beslutet ha bifallits med minst 2/3 majoritet.

Kallelse till stämma som behandlar föreningens upplösning skall innehålla ett av initiativtagarna - styrelsen, revisorerna eller grupp av medlemmar om minst 30 - motiverat förslag till nedläggningsbeslut med ett tydligt förslag om fördelningen av föreningens tillgångar.

ÅRSMÖTES PROTOKOLL

FÖRENINGEN PROGRAMBITEN
Stämman

ÅRSMÖTESPROTOKOLL
1985-02-23

Mötet ägde rum i Militärhögskolans samlingssal i Stockholm den 23 februari 1984 med början kl 1500

1. Mötet öppnades av föreningens ordförande Lennart Lindberg. De närvarande kontrollerades mot medlemsregistret av Claes Schibler.

2. Till ordförande för mötet valdes Lennart Lindberg, till sekreterare Bengt Fahlgren och till justeringsmän Per Rundqvist och Jan Alexandersson.

3. Mötet förklarades vara behörigen utlyst.

4. Dagordningen godkändes sådan den presenterats i kallelsen.

5. Styrelsens årsredovisning 1984.
- Verksamhetsberättelsen presenterades

- Kassarapporten presenterades i form av en resultaträkning och en balansräkning.

Mötet beslöt att lägga årsredovisningen till handlingarna.

6. Lars Hedlund föredrog revisorernas berättelse. Revisorerna föreslog att styrelsen för 1984 skulle beviljas ansvarsfrihet.

7. Fråga om ansvarsfrihet för styrelsen.
Gösta Blume övertog ledningen av mötet tillfälligt. Styrelsen beviljades ansvarsfrihet för sin förvaltning sådan den framgått av lämnad redovisning.

8. Styrelsens förslag till reviderade stadgar. Stadgarna diskuterades utifrån den version som presenterats i kallelsen. Diskussionen rörde främst paragraf 1 rörande föreningens syfte. Mötet beslöt att föreningen skall vara öppen för alla men att tidningen främst skall behandla TI-99/4A och TI-59, möjligen andra räknare. Beslut om ändring av denna inriktning skall tas på föreningsstämma. Vissa andra förslag av teknisk och språkmässig art framfördes av Gösta Blume. Mötet uppdrog åt styrelsen att med hjälp av Gösta Blume göra erforderliga ändringar i texten. Mötet beslöt att de sålunda reviderade stadgarna skulle antagas. Detta innebär att de skall tillämpas under verksamhetsåret 1985 i den mån de ej strider mot de äldre stadgarna samt att de skall tas upp på en senare stämma för eventuell fastställelse.

9. Verksamheten 1985.
Verksamheten 1985 diskuterades. Den bör enligt mötets mening innehålla bl a följande:
Tidning, fyra nummer
Medlemsmöten, två stycken
Special Interest Groups, någon slags aktivitet
Utgivning av matrikel
Värkning av nya medlemmar
Internationella kontakter
Presentationsbroschyr
Annonsering om föreningen

10. Val av styrelse för 1985.
Valberedningen föredrog sitt förslag. Därefter valdes - till ordförande Lennart Lindberg
- till övriga styrelseledamöter Claes Schibler, Göran Nygren, Tony Rogvall, Michael Öhman, Börje Håll och Jan Alexandersson. Årsmötet uttryckte även sin önskan att Gösta Blume skulle bli kassör. Om Blume efter betänketid accepterade uppdraget skulle han anses vald. Detta skulle i så fall bekräftas av styrelsen, som i annat fall hade att utse annan kassör. Det beslöt att föreningens medel skall disponeras av Lennart Lindberg samt den blivande kassören. Teckningsrätten för postgirokonton 19 83 00-6 samt 430 01 59-3 gavs till Lennart Lindberg och den blivande kassören var för sig. (Gösta Blume har sedermera accepterat kassörsuppdraget /Protokollföraren)

11. Val av revisorer.
Till revisorer valdes Bo Carleö och Mikael Nordlin.

12. Val av valnämnd.
Till ledamot av valnämnden valdes Anders Thörnqvist. Han utlovades assistans av Göran Nygren, som bedömdes ha lämpliga kontakter.

13. Årsavgiften för 1985 fastställdes till 120:- kronor.

14. Inga beslut fattades under denna punkt.

15. Inga beslut fattades under denna punkt.

16. Övriga frågor.
Föreningen har från USA fått tillgång till följande program, som kan användas fritt:

Källkoden till TI-FORTH version 3.0
Ny version av TI-MULTIPLAN och TI-WRITER
En ny version av TI-DEBUGGER

Programdistribueringen diskuterades mot bakgrund av en motion från Lars Erik Swahn. Den innebar att kopiering och sändning av beställda program skulle spridas på flera händer.
Motionen avslogs sedan styrelsen förklarat sig kunna förbättra servicen med nuvarande organisation och menat att problemen med en spridning kunde förväntas uppväga fördelarna.

17. Mötet avslutades kl 18.

Vid protokollet

Bengt Fahlgren
Bengt Fahlgren

Justeras

Lennart Lindberg
Lennart Lindberg

Per Rundqvist
Per Rundqvist

Jan Alexandersson
Jan Alexandersson

ÅRSBERÄTTELSE

Styrelsens årsredovisning för 1984.

Inledning

Verksamhetsåret 1984 inleddes i en stämning, präglad av Texas Instruments beslut att dra sig ur hemdatormarknaden och lägga ned tillverkning och försäljning av TI 99/4A. Vad som hände visade snabbt att det inte var fel på maskinen utan snarare på marknadsföringen. Tredje-parts-tillverkare dök snabbt upp, programutbudet ökade och användarföreningarna världen över visade stor aktivitet.

PROGRAMBITENS verksamhet kring TI 99 har under året varit tämligen livlig. Medlemsantalet har ökat. Intresset för de programmerbara räknarna har funnits kvar.

Styrelse och funktionärer.

Föreningens angelägenheter har under det gångna året handhåfts av en styrelse med följande sammansättning:

Lennart Lindberg, ordförande
Bengt Fahlgren, kassör
Claes Schibler, registerförare, materielförvaltare mm
Göran Nygren, redaktör
Michael Öhman, biträdande redaktör
Tony Rogvall
Martin Holmberg
Hans Attersjö, programförmedlare, som dock avflyttade utrikes i maj och då ersattes av
Björn Mårtensson

Barbro Nygren valdes på årsmöte till sekreterare men avsåg sig snabbt av personliga skäl. Sysslan har skötts av Lennart Lindberg.

Björn Gustavsson och Bo Nordlin har vid några tillfällen biträtt styrelsen.

Föreningens FORTH distribueras av Örjan Gustavsson i Smedjebacken.

Anders Persson i Lund har varit redaktör för Utmaningen i tidningen bl a. Såväl han som Peter Odelryd har bidragit med utlandskontakter.

Under året har hållits nio protokollförda styrelsemöten. Ett tionde planeras till dagen för årsmötet.

Föreningens ekonomi.

Föreningen har under året haft en god ekonomi. Till detta har starkt bidragit en återhållsamhet med ekonomiska åtaganden i allmänhet och framförallt den kraftigt sänkta kostnaden för produktion av tidningen, som åstadkommits dels genom en enkel tryckteknik, dels genom inskrivning, utskrift och montering med frivilliga krafter. Claes Schiblers kunskaper och insatser har här varit av stort värde.

Arvoden till funktionärer har inte utgått.

Föreningens inkomster, utgifter, tillgångar och skulder framgår av bifogade räkenskapshandlingar, innehållande en resultaträkning med en omslutning av 100 947:72 kronor och en balansräkning omslutande 63 745:68 kronor. Den uppkomna vinsten, 26 495:08 kronor, föreslås föras i ny räkning.

Föreningens verksamhet.

Fyra planerade nummer av tidningen PROGRAMBITEN NITTINIAN har givits ut. En medlemsmatrikel har distribuerats.

Två allmänna medlemsmöten har hållits i Stockholm under hösten och ett i Lund. De har varit välbesökta - i Stockholm ett sextiototal personer per gång - och lyckade.

En omfattande korrespondens och telefonrådgivning har pågått mellan styrelse och andra funktionärer å ena sidan och medlemmar i allmänhet å den andra.

Programbanken har reorganiserats och rensats. Den har ökat i omfattning liksom förmedlingen av program.

Ett nät av internationella kontakter har börjat byggas upp, varigenom medlemmarna får tillgång till information och program.

Försäljningen av äldre tidningsnummer, FORTH mm har haft en omslutning av ca 25 000:- kronor.

Att förbättra

Den systematiska medlemsvärningen bör öka. Det finns troligen en betydande potential i Norden för utökningar, som bör gagna alla.

Det finns berättigade önskemål om att föreningens funktionärer reagerar snabbare på brev och beställningar.

Stockholm datum som ovan.
För styrelsen

Lennart Lindberg, ordförande Bengt Fahlgren, kassör

FÖRENINGEN PROGRAMBITEN

RESULTAT OCH BALANSRÄKNING 1984

INGÅENDE BALANS 1/1 1984

TILLGÅNGAR:		SKULDER OCH EGET KAPITAL:	
KASSA	661.55	OBET. FAKTUROR	50301.10
POSTGIRO	67771.16	SKULD MEDL. 1984	6160.00
BANK	0.00	SKATT 1983	402.00
VARULAGER	1500.00	SKULD SKATT 1982	889.00
INVENTARIER	2700.00	ARB.LÖN (PREL SKATT)	2273.00
FORDR. RÄNTOR	1106.45	UTG. MOMS	8303.00
FORDR. VAROR	333.35	TID. UTFÄSTELSER	7000.00
FORDR. MEDLEMSAVG.	200.00	EGET KAPITAL	8129.41
ING. MOMS	9185.00		
SUMMA:	83457.51	SUMMA:	83457.51

RESULTATRÄKNING FÖR 1984

KOSTNADER:

TRYCK TIDNING 4 NR	26934.00
VARUINKÖP	23948.96
PORTO	9794.94
DIV. OMKOSTNADER	6424.74
AVSKRIVNING INVENTARIER	540.00
SKATTER	2400.00
LOKALER	4410.00
ÅRETS VINST	26495.08
SUMMA	100947.72

INTÄKTER:

MEMLEMSAVGIFTER	59183.74
VARUFÖRSÄLJNING	28347.65
ANNONSER	3070.50
RÄNTOR	3389.83
FÖR HÖGT UPPTAGNA SKULDER	6956.00
SUMMA	100947.72

UTGÅENDE BALANS 31/12 1984

TILLGÅNGAR:		SKULDER OCH EGET KAPITAL:	
KASSA	322.40	OBET. FAKTUROR	20372.60
POSTGIRO	54874.03	SKULD MEDL 1985	1180.00
BANK	0.00	PREL. B-SKATT 1984	490.00
VARULAGER	1500.00	UTG. MOMS	7078.59
INVENTARIER	2160.00	EGET KAPITAL	8129.41
FORDR.ARB.GIV.AVG.	1189.00	ÅRETS VINST	26495.08
ING. MOMS	3700.25		
SUMMA:	63745.68	SUMMA:	63745.68

STOCKHOLM 13/2 1985

Bengt Fahlgren
BENGT FAHLGREN / KASSÖR

UTMANINGEN!

Redaktörens adress:
Anders Persson
Kännärsvägen 4:1078
222 45 LUND

Jaha, då inleder vi ett nytt år med Programbiten. Förhoppningsvis med friska tag både här och där.

TELEFONKONFERENS

Det påstås att en telefonjour ska startas i Stockholm, dit man kan ringa om man vill fråga om något. Eftersom det tar ganska lång tid från det att ett brev kommer till mig och fram till en lösning i tidningen, har jag redan i praktiken startat en telefonservice. I de fall jag har ansett samtalen ha ett allmänt intresse har de givit upphov till artiklar i Utmaningen.

Jag är inte hemma så mycket, men numret är 046-116919 om någon vill försöka.

MEDLEMSTRÄFF?

Det har redan förekommit en liten rapport i tidningen angående vad vi pratade om på medlemsträffen i Lund, men jag vill ändå återkomma till ämnet. Eftersom det är uppenbart att det finns ett intresse för, eller kanske snarare behov av, att träffas är upmaningen (Utmaningen!) given: Ordna en träff! Det är inte alls svårt. Kan du ordna en lokal går resten nästan av sig självt. I Lund kom det nästan 40 personer, i Stockholm brukar det vara sådär 70-80, sägs det.

Hur man vill lägga upp det är naturligtvis fritt för var och en, men jag tror att man bör ha ett visst tema, som tar upp en del av tiden. Annars blir det lätt bara rundsnack, och ingen lär sig något. Om någon tänker ordna något i södra ändan av landet går det bra att ta kontakt med mig för tips.

MERGE FRÅN KASSETT (50)

På medlemsträffen i Lund diskuterade vi återigen merge med kassetbandspelare. Intresset för detta är tydligen ganska stort. Om man till att börja med begränsar sig till Extended BASIC, ser man att minnesexpansionen är absolut nödvändig. Då kan man lagra ett merge-program i 8k-delen av minnet. Den delen av minnet lämnas orörd av BASIC, utom när man gör en CALL INIT. Enklarest är nog att mer eller mindre imitera den metod för merge som TI använder i X-BASIC. Den går i korthet ut på att program som ska slås ihop med andra program lagras på ett sätt som är speciellt anpassat för det ändamålet. Genom att skriva ett assemblerprogram med två funktioner kan man uppnå detta.

Den ena funktionen läser av programmet som finns i minnet, gör om det till ett format som passar till merge och lagrar det på angiven fil, t.ex. CS1.

Den andra funktionen läser in ett sådant specialprogram, samsorterar det med det program som redan finns i minnet och ger på så vis programmen hopslagna till ett.

För att kunna göra detta måste man veta hur raderna lagras i minnet. Nu förutsetter jag ju att minnesexpansionen är ansluten, men principen är den samma även om man bara har grundenheten. Det är av intresse när det gäller att skriva en variant som fungerar med TI-BASIC och Mini Memory.

Raderna lagras i byteformat. De kan alltså börja och sluta på både jämna och udda adresser. Den första byten anger radens längd. Om man tar adressen där den byten finns, och lägger till längden som byten anger, får man adressen till radavdelaren. Efter längdangivelsen kommer själva raden. Den är kodad på ett speciellt sätt, men det har ingen betydelse för det här programmet. Som exempel kan nämnas att vanliga instruktioner (PRINT, INPUT, RETURN osv) alla lagras som en byte i minnet. Koderna mellan 128 och 255 används till detta. Instruktionerna i exemplet lagras som 156, 146 och 136. Efter varje rad kommer en radavdelare, vilken har koden noll (0).

Raderna kan komma i vilken ordning som helst. Ordningen beror på hur de matades in. Den raden som matades in eller ändrades sist kommer först i minnet (lägst adress). Lägg också märke till att radernas nummer inte lagras tillsammans med raderna själva.

Hur kan då datorn veta i vilken ordning den ska köra programmet? Jo, den har en särskild tabell, radnummertabellen, som anger detta. I den tabellen har varje rad fyra bytes. De två första anger radens nummer. Radnummertabellen är sorterad så att det högsta radnumret kommer först i minnet (lägst adress). Nästa två bytes talar om vilken adress som raden i fråga faktiskt finns på. Adressen man får här pekar på den byte som kommer omedelbart efter den som anger radens längd.

Nu räcker det inte med detta. Datorn måste veta var de olika tabellerna finns i minnet också. Det håller den reda på genom diverse pekare.

8330 (hex) innehåller startadressen till radnummertabellen, dvs var information om programmets sista rad är lagrad. -31952

8332 (hex) anger var radnummertabellen slutar. -31950
8384 (hex) är den högsta adressen i 24k-delen som man kan använda.

Eftersom radnummertabellen och programraderna alltid ligger intill varandra, börjar programraderna på adressen (8332)+1. Parentesen anger att det är innehållet i 8332 som avses.

Filstruktur.

Om man nu låter ett assemblerprogram läsa av dessa tabeller kan man få programmet i ett annorlunda format. I början av raden lagras radnumret och det antal bytes som raden upptar. Sedan kommer själva raden. På detta sätt eliminerar man behovet av någon radnummertabell. Alla raderna lagras efter varandra i en filbuffert i VDP RAM. När hela programmet bearbetats på detta sätt, skriver man ut det på önskad fil. Filtypen skall naturligtvis vara "program". Det är det enda format som tillåter en hjälpligt effektiv lagring på kassett. Lämpligen låter man filnamnet vara en parameter till assemblerprogrammet, så att man kan utnyttja programmet även med disketter. Det blir enklare att testa då.

När "merge" beordras läses filen in. Sedan börjar man stoppa in respektive radnummer i den radnummertabell som redan finns i minnet. Även en tom tabell, dvs inget program i minnet, ska tillätas. Om samma radnummer finns i båda programmen, ska den nya raden ersätta den befintliga. Raderna själva måste naturligtvis också stoppas in i minnet. När man har gjort detta med alla raderna, är det klart! Lätt, va?

För att det ska fungera i praktiken är det nog bäst att lagra någon identifieringstext i början av filen. Om man försöker sig på att slå ihop ett program med en fil som inte är avsedd för detta blir det kaos annars. Dessutom behöver man någon form av filslutsmarkering, för att veta när man ska sluta.

Den här principen, med filtypen "program", gör naturligtvis att man inte kan använda merge med alltför stora program. Hela filen måste ju läsas in på en gång. Det är antagligen därför Texas Instruments valde att endast tillåta merge från diskett. Där kan man ju hyfsat snabbt läsa filer som är uppbyggda med många poster.

Metoden ovan gäller alltså Extended BASIC när minnesexpansionen finns. Om man inte har extra minne finns det ingen plats för det assemblerprogram som behövs. Däremot kan man utnyttja Mini Memory om man har den modulen. Då kan man få en merge-möjlighet för program skrivna i både TI och Extended BASIC. Det blir väl lite värre att få plats med allt som behövs i Mini Memory RAM, men det går förhoppningsvis. Dock måste man ta med i beräkningen den krångliga minneshanteringen som uppstår när man bara har VDP RAM att tillgå. Bl.a. av den anledningen anser jag att det är lämpligt att börja med X-BASIC och 32k extra RAM. När det fungerar kan man utveckla andra versioner.

GENERELL PROGRAMSTARTARE (46).

Det här är ett ganska gammalt problem. Det har tidigare diskuterats i 83-4/5 och 84-1. I nummer 84-3 fanns dessutom en lösning utanför Utmaningen, men ingen förklaring till hur den skulle användas.

Bengt Fahlgren har skickat in en annan variant, som är lite mer användarvänlig. Den bygger på samma princip, dvs att filnamnet i RUN-satsen förändras med CALL LOAD. Det gör att Extended BASIC tillsammans med minnesexpansionen krävs.

Bengts program visar endast filer som är lagrade i "program" format på skivan. De flesta datafiler sorteras alltså bort. Detta är bra för det mesta, men ger den bieffekten att stora program inte kan startas, eftersom de lagras med filformatet Internal Variable 254. Med stora program avses (vanligen) större än 11584 bytes. Å andra sidan finns det ingen möjlighet att bara genom att titta i filkatalogen avgöra om en sådan fil verkligen är ett BASIC program eller ej.

Eftersom programmet ska modifiera sig självt är det känsligt för var det finns i minnet. Om man matar in rad 670 först av alla, blir den kvar på ett ställe i minnet, även om man förändrar de andra raderna. Då behöver man inte ändra på adresserna i CALL LOAD satserna.

För att ta reda på var rad 670 finns i minnet kan man skriva in följande kommando:

```
FOR I=1 TO -24576 STEP -1 :: CALL PEEK(I,A) :: PRINT I,CHR$(A) :: NEXT I
```

Då får man med tiden se något i den här stilen:

```
-293 *
-294 *
-295 *
-296 *
-297 *
-298 *
-299 *
-300 *
-301 *
-302 *
-303 .
-304 1
-305 K
-306 S
-307 D DADR
```

Tryck på FCTN-4 när det här visar sig. Med ledning av detta kan du placera in de rätta adresserna på raderna 630, 650 och 660.

```
630 CALL LOAD(DADR+4,
650 CALL LOAD(DADR-1,
660 CALL LOAD(DADR+3,
```

Vad DADR blir i ditt fall är alltså allt du behöver ta reda på. I det publicerade programmet (i slutet av Utmaningen) är DADR = -397. Om du knappar in programmet själv får du nästan garanterat ett annat värde.

Vid körning av programmet får man en numrerad lista med programmen på den diskett som anges. Sista numret (40) ger återstart, om man inser att det var fel diskett.

Det här programmet kan man lagra som DSK1.LOAD på de disketter som innehåller körklara BASIC-program.

Det här är Bengts programstartare.

Extended BASIC, minnesexpansion och disk.

```
100 !*****
110 !* GENERAL PURPOSE *
120 !* PROGRAM LOADER *
130 !*****
140 !* ORIGINAL PROGRAM BY *
150 !* <---?????????---> *
160 !* THIS ENHANCED AND *
170 !* CHANGED VERSION WAS *
180 !* CREATED BY: *
190 !*
200 !*--- BENGT FAHLGREN ---*
210 !*
220 !* MARCH 13,1984 *
230 !* STOCKHOLM SWEDEN *
240 !*****
250 !* WARNING!!!! *
260 !* CHANGES IN THIS *
270 !* PROGRAM MIGHT AFFECT *
280 !* THE SPECIFIC LOAD *
290 !* ADDRESS IN LINES *
300 !* 630,650 AND 660 *
310 !*****
320 CALL INIT :: CALL LOAD(-31878,0)
330 CALL CLEAR :: CALL SCREEN(8):: FOR C=0 TO 14 :: C
ALL COLOR(C,2,8):: NEXT C
340 CALL CHAR(96,"0000FFFF")
350 OPTION BASE 1
360 DIM PG$(40)
370 DISPLAY AT(6,10):"DISK MENU" :: DISPLAY AT(7,1):R
PT$(",",28)
380 DISPLAY AT(12,4):"WHICH DISK? (1-3) 1";
390 ACCEPT AT(12,24)SIZE(-1)VALIDATE("123"):D$
400 OPEN $1:"DSK"&D$&".",INPUT,RELATIVE,INTERNAL
410 INPUT $1:N$,A,A,A
420 DISPLAY AT(1,6)ERASE ALL:"DSK"&D$&" - "&N$;
430 DISPLAY AT(2,1):RPT$(" ",28):: I=0
440 DISPLAY AT(13,8):"PLEASE WAIT !"
450 FOR X=1 TO 39 :: I=I+1 :: IF I>127 THEN 630
460 INPUT $1:P$,A,B,B
470 IF LEN(P$)=0 THEN 500
480 IF ABS(A)>5 THEN 460
490 PG$(X)=P$ :: NEXT X
500 DISPLAY AT(13,7):RPT$(" ",15)
510 FOR J=1 TO 20
520 DISPLAY AT(J+2,1):USING "$$ $$$$$$$$ $ $$$$$$
$$$$$:J,PG$(J),J+20,PG$(J+20)
530 NEXT J
540 DISPLAY AT(22,16)SIZE(-15):"40 restart" :: DISPLA
Y AT(23,1):RPT$(" ",28)
550 DISPLAY AT(24,1):" YOUR CHOICE? 1"
560 ACCEPT AT(24,25)SIZE(-2)VALIDATE(DIGIT):K
570 IF K=40 THEN 680
580 IF K<1 OR K>39 THEN 550
590 IF LEN(PG$(K))=0 THEN 550
600 CLOSE $1
610 FOR P=1 TO LEN(PG$(K))
620 A$=SEG$(PG$(K),P,1)
630 CALL LOAD(-393+P,ASC(A$))
640 NEXT P
650 CALL LOAD(-398,5+LEN(PG$(K))):: CALL LOAD(-393+P,
0)
660 CALL LOAD(-394,ASC(D$))
670 RUN "DSK1.*****"
680 CLOSE $1 :: CALL CLEAR :: GOTO 370
```


51. DISKETTRUTINER

Bengt Fahlgren vill gärna ha ett assemblerprogram som kopierar disketter. Samma som Disk Manager alltså, men gärna fortare. Dessutom skulle man kunna skriva ett intelligentare program som eliminerar en hel del av det nödvändiga diskettskyfflandet om man bara har en drivenhet.

Bengt vill också ha ett diskatalogprogram i assembler. Möjlighet att få ut katalogen både på skärm och skrivare måste förstås finnas.

Det här är inte alls så onödigt som det kanske verkar, med tanke på att det redan finns en Disk Manager modul. Om man skriver de här programmen så att de kan anropas från Extended BASIC, kan man ladda in den här utbyggnaden av DOS:en (Disk Operativ Systemet) i 8-k delen av minnesexpansionen. Då finns det alltid tillgängligt, utan att byta modul och åka ur X-BASIC. Även om man använder TI BASIC måste man ju hoppa ur BASIC innan man kan komma in i Disk Manager. Om man då har Editor/Assembler eller Mini Memory kan man uppnå samma funktion där som för X-BASIC.

Att tanken är vettig visas också av att det i en broschyr, som jag fick av Maurice Swinnen, finns ett program som kallas Disk Manager III. Det gör precis det jag beskrivit ovan, och kan dessutom formatera disketter, ändra fil- och disknamn, kopiera osv.

Det finns ett antal subprogram inbyggda i diskkontrollkortet. FILES är ett av dem, men det finns andra också. Bland dessa finns program för formatering av en diskett och för att läsa eller skriva en sektor, har jag listat ut. Men det finns fler. Kanske någon annan har forskat i vad de egentligen gör?

52. VIDEOSIGNALER M.M.

Både Lennart Thelander och Bengt vill ha en "låda" som gör att man kan utnyttja videoingången på en TV eller monitor. Utsignalen från datorn i den version som vi använder här i Sverige består ju av videosignal med separat färginformation. Om man då vill ha en bild på en färgmonitor kan man inte bara stoppa in videosignalen i den. Gör man det blir det svart/vitt.

Antingen kan man tänka sig en konstruktion som kopplas till videoutgången på datorn, eller också en modifiering av RF-modulatern. Rimligen bör ju en videosignal med färginformation finnas i den, innan den görs om till antensignal.

Det finns ju också SCART eller EURO-kontakter på en del moderna apparater. Dessa har då ibland även RGB-ingång. Någon sådan signal ger inte 99:an ifrån sig, men det skulle man ju kunna åstadkomma, det också.

Någon som har experimenterat med detta och kan dela med sig av erfarenheterna?

Aha! Vidare forskning på området har visat att TISOFT har schema på lite av varje, däribland en "RGB ENCODER". Dessutom finns en ritning på RF-modulatern.

53. TRACE PÅ ASSEMBLERPROGRAM

Folke Andersson har sett ett program till en VAX 11/780 som visade hur data flyttades omkring i maskinen när den exekverade ett program. När Folke fick se den nya debuggern med enkelstegning som Navarone tagit fram, kom han att tänka på Digitals uppvisning. Varför inte göra något liknande på 99:an?

Jag blev själv intresserad av id'n, och har funderat en del på hur det skulle kunna göras.

Enligt Folke kunde man på VAX:en se precis hur talen flyttades mellan olika register inom CPU:n och till och från minnet. Jag anser inte att det är detta som är mest intressant. De som kan begripa det förstår ändå hur det fungerar utan någon uppvisning på

skärmen. Dessutom behöver man känna till hur maskinen är mikroprogrammerad för att kunna få det rätt. Det är kanske inte så lätt att få reda på. Däremot tänkte jag att man skulle kunna göra ett program som visar hur innehållet i processorns register förändras efter varje instruktion. Med register menar jag då WP, PC, ST och de 16 register som ingår i "workspace". Dessutom kan man tänka sig att även övervaka några intressanta minnesadresser på skärmen.

Vad ska man då ha det här till? Ja, som utvecklingshjälpmedel är det väl knappast användbart. Då fungerar en vanlig debugger bättre, särskilt som den nya "Super-bugger" kan enkelstega program också. Men när någon ska lära sig assemblerprogrammering, då vore det mycket praktiskt om man på detta sätt verkligen kunde se vad som händer i maskinen. TMS 9900 har ju en del ovanligheter för sig med arbetsareor och liknande. Dessa skulle man mycket lätt kunna visa med hjälp av ett sådant här program.

Hur då implementera härligheten? Själv funderade jag på att göra en virtuell maskin, dvs ett program som simulerar en TMS 9900. Detta program kan då skrivas i vilket högnivåspråk som helst. Forth t.ex. Det är alltså inte frågan om att övervaka ett speciellt assemblerprogram, utan interpretera instruktioner som ingår i processorns språk.

Genom att göra så här kan man också kringgå de begränsningar som har lagts på TMS 9900 i just TI 99/4A. Man kan exempelvis simulera processorns hela avbrottsystem, i stället för de ganska begränsade möjligheter som finns kvar i 99:an.

För den särskilt intresserade kan nämnas att möjligheterna på 99/4A är ungefärligen jämförbara med de på andra datorer i samma klass, men inte alls så avancerade som mikroprocessorn egentligen tillåter. Den med tanke på åldern tämligen avancerade avbrotts hanteringen som TMS 9900 har, med hårdvarumässig avkodning av 16 olika prioritetsnivåer, hänger samman med att processorn, när den kom 1976, främst var tänkt som en styrprocessor. Därav det speciella systemet med 16 register, placerade på godtycklig plats i minnet. Eftersom varje avbrott leder till byte av register kan man lätt hoppa mellan olika arbetsuppgifter (context switching). Något som ofta förekommer vid styrning av maskiner och liknande i realtid.

I en simulator skulle man kunna känna av tangentbordet och låta vissa tangenter betyda avbrott med olika prioritet.

9900 kan ju adressera 64K bytes minne, vilket är mer än vad som finns som RAM i de flesta 99:or. Men det kan man klara med ett slags virtuell minneshantering. Det är lätt ordnat genom att använda 64K bytes av en diskett till att simulera minnet och läsa in de delar som behövs för tillfället. En annan metod är att begränsa det tillgängliga minnet, så att det får plats i datorn.

BASIC är något olämpligt till det här, bland annat beroende på att heltal saknas. Forth eller Pascal kan däremot med fördel användas. Dessutom kan man naturligtvis skriva simulatoren i just assembler, men då är det nog inte gjort i en handvändning. Det är det kanske inte ändå, för den delen.

Hur programmen ska matas in till simulatoren får vara upp till den som försöker sig på det här att avgöra.

54. LOCATE I FORTH

Från Lennart Lindberg kommer förslaget att åstadkomma en LOCATE-funktion till TI-Forth. På sidan 245 i Brodie's Starting Forth finns en beskrivning på hur det ska fungera. Det gäller alltså ett ord som letar upp vilken skärm på disketten som ett angivet ord är laddat ifrån.

Villkoret är att ordet ifråga ska finnas inladdat när LOCATE används, samt att disketten med skärmen det kommer ifrån verkligen sitter i maskinen.

Lennart har egna tankar om hur det här ska se ut, vilka jag visar dem sist i den här artikeln. OBS! Skrivet för TI-Forth 3.0

På skärmarna finns några ord som inte är definierade här, men de ska Lennart ha beskrivit i "Årsmötesnumret", vilket inte är utkommet när detta skrives. Definitionerna är följande:

: FH BLK + ;

: THRU 1+ SWAP DO I LOAD LOOP ;

: Ø IN 64 / 1+

64 * IN ! ; IMMEDIATE

Det sista ordet lider av den evinnerliga plågan med konstiga europeiska tecken. Skriver jag nu att det är Ø, inte Ö, blir ingen klokare av det. Se det så här: Det sista ordet består av ett tecken, vilket är ett bakåtlutande divisionstecken, inte den sista bokstaven i svenska alfabetet.

Ordet i fråga förekommer flitigt på Lennarts skärmar, men när jag skriver det här har jag ingen aning om hur det kommer att ta sig ut i tidningen. Suck...

UNDERPROGRAM

De av mina sju läsare, eller hur många det nu är, som orkade igenom Utmaningen sist, när det handlade om underprogram och telefonregister, har kanske tankar om detta. Särskilt då registerprogrammet, eftersom man kan ha många meningar om hur det bör och inte bör se ut. Kom gärna med nya varianter på hela eller delar av det programmet.

För den som läste mycket noga kan jag meddela att sista siffran i PI ska vara 6 (sex), inte 9 (nio). Dessutom är arean för en cirkel nämnda PI gånger radien i kvadrat. Skrivaren i Stockholm behandlade inte texten likadant som min gör.

Dessutom finns det (minst) ett fel i miniregistret. Det kan ordnas till genom att rad 610 får se ut så här:

```
IF POS("JJ",SVAR$,1) AND ANTAL<MAXANTAL THEN PRINT ::  
GOTO 520
```

Annars blir registret inte sorterat om man uppnår maximal kapacitet vad det gäller antalet poster.

Knepet med POS ovan är ganska praktiskt när man ska kolla om en bokstav stämmer överens med ett antal olika varianter. Speciellt i TI BASIC, som saknar ACCEPT med VALIDATE samt OR, AND och liknande nyttigheter. Jämför själv:

```
IF (FLER$<>"J") + (FLER$<>"j") + (FLER$<>"N") +  
(FLER$<>"n") =-4 THEN 200
```

```
IF POS("JjNn",FLER$,1)=0 THEN 200
```

Exemplet kommer från P:TEXTUT i nummer 84-3. Strängt taget är de här raderna inte likvärdiga, eftersom den andra varianten inte säkert fungerar om man matar in mer än en bokstav som svar. Någon med beslutsvända kan säkert få för sig att svara med "jN", om inte annat så för att se vad som händer. Men man kan ju hugga av resterande tecken med

```
FLER$ = SEG$(FLER$,1,1)
```

Det här felet har jag hittat själv, men den förste "utomstående" som påpekar ett eventuellt fel lovar jag framföra ett speciellt tack till. Låt det bli en Utmaning!

47. SORTERING

I 84-2 efterlyste jag program för att sortera strängar. Något sådant har inte dykt upp, men däremot har Bengt Fahlgren önskat sig ett dylikt som klarar av att få rätt på svenskan.

Som de flesta väl känner till har vi ju problemet med att ASCII koden påstår att Å och Ö kommer före Ä. Det här kan man åtgärda till exempel genom att lägga in särskilda test vid jämförelsen av två strängar. Dessa test får då avgöra om den vanliga ASCII tabellen kan användas, eller om speciella åtgärder måste tillgripas.

En annan metod är att före sorteringen byta ut alla Å, Ä och Ö mot något annat, som ger rätt ordning. Efter sorteringen måste man då byta tillbaka igen. Allt det här kan naturligtvis göras i BASIC, men det är arbetsamt för datorn och tar naturligtvis en förskräcklig tid. Assembler, assembler...

Det vore intressant att få höra av någon som har anfallit det här problemet med sortering på svenska tidigare.

KÜPES

32K Fristående minnesexpansion till TI-99/4A

Måns Tostéberg
Opalstigen 13
282 00 Tyringe

Tel 0451/51192

LISTA ORD

av Jan Alexandersson

Jan skickar en hel del av sina bidrag på skiva, inskrivna med Ed/Ass/modulens editor. För att underlätta redaktions arbete - kanske också sitt eget - har han gjort ett program som listar dessa filer i ett format lämpat för tidningen. Programmet kollar att raderna inte är över 55 tecken långa (det blir en spalt, det, när man listar med 12 tecken per rad)./Red

```
100 REM LISTA ORD XB  
110 REM JAN ALEXANDERSSON  
120 REM VERSION 1985-01-22  
130 CALL CLEAR  
140 CALL CHAR(91,"00280038447C44440028007C4444447C003  
82838447C4444")  
150 CALL CHAR(123,"0000280038447C44000028007C44447C00  
00382838447C44")  
160 CALL COLOR(13,7,7,14,13,13)  
170 PRINT "Listningen kan:"stoppas med valfri tangen  
t, startas med valfri tangent": :  
180 INPUT "ÖNSKAD FIL ":FIL$  
190 GOSUB 390  
200 OPEN $1:FIL$,INPUT  
210 LINPUT $1:TEXT$  
220 PRINT TEXT$  
230 BYTES=BYTES+LEN(TEXT$)  
240 RAD=RAD+1  
250 IF LEN(TEXT$)>55 THEN PRINT RPT$(CHR$(128),10);"V  
ARNING LÄNGD=";LEN(TEXT$): CALL SOUND(50,440,0):  
: FEL$=FEL$&STR$(RAD)&" "  
260 CALL KEY(3,KEY,STA): IF STA<>0 THEN GOSUB 350  
270 IF EOF(1)=0 THEN 210  
280 CLOSE $1  
290 GOSUB 350  
300 GOSUB 390  
310 PRINT "BYTES=";BYTES,"RADER=";RAD  
320 IF FEL$<>"" THEN PRINT RPT$(CHR$(128),28);"FÖLJAN  
DE RADER ÄR FÖR LÅNGA":FEL$330 GOSUB 350  
340 END  
350 REM PAUS  
360 CALL KEY(3,KEY,STA)  
370 IF STA<1 THEN 360  
380 RETURN  
390 REM GRÖN LINJE  
400 PRINT RPT$(CHR$(136),28)  
410 RETURN
```

SCR #33

```

0 ( Laddskärm för sök ord i SCREEN )
1 ." obs! 3DUP och UNDER+ " CR
2 1 FH 4 FH THRU ;S
3 Med JMF$ kan man söka en kort sträng i en längre.
4 Med SEARCH kan JMF$ användas för mera organiserad sökning.
5 SEARCH-PREP förbereder för SEARCH och SEARPR registrerar
6 resultatet. Är ordet funnet stannar sökningen i Editorn.
7 FINN letar i ordlistan.
8 LETA klarar ut sökningen med sökordet i PAD.
9 LET.BAS innehåller startadress på blocket man letar i.
10 BLEDDRA plockar fram nya block.
11 BLEDScreen innehåller blocknumret där sökning börjar.
12 LOCATE är toppordet. När ordet är funnet hamnar man i Editorn.
13 FCTN 9 lämnar Editorn som vanligt.
14 P <ENTER> fortsätter sökningen där den stoppade.
15

```

SCR #34

```

0 ( Sök ord i screen, 1 av 4 )
1 0 VARIABLE LET.BAS 0 håller blockets startadress för sökstart
2 : JMF$ ( a1 n a2--?) 0 En byte-variant av -TEXT SF sid 282
3 0 a1=adr sökord, a2=adr basstr, n=ant tkn att undersöka(=sökord
4 0 ?=0 lika, ?=1 sökord>basstr, ?=-1 sökord<basstr
5 OVER OVER + SWAP 0 Sätt ramar i basstr för sökloop
6 DO DROP 0 starta loop, drop n
7 1+ 0 skapa startvärde för nästa varv
8 DUP 1- C$ 0 hämta en byte fr sökordet
9 I C$ 0 hämta en byte fr bassträngen
10 - DUP 0 jmf med subtraktion. Dup flaggan
11 IF DUP 0 dup om skilt fr 0, dvs de är olika
12 ABS / 0 dividera med absolutvärdet. Ger 1 eller -1
13 LEAVE 0 lämna vid nästa LOOP då de är olika
14 THEN LOOP SWAP DROP ; 0 drop nästa adress, lämna flaggan
15

```

SCR #35

```

0 ( Sök ord i screen, 2 av 4 )
1 0 VARIABLE BLEDScreen 0 håller blocknumret där sökning startar
2 : SEARCH-PREP ( a1 a2--a1+1 n1 a2 1 1024)
3 >R COUNT 0 flytta undan a2, dela på a1
4 1 R> 1024 0 lägg en flagga, in m a2, lägg längden a2
5 ROT SWAP ; 0 upp m flaggan, upp m längden a2
6
7 : .PAD PAD COUNT TYPE ;
8 : SEARPR ( n ?-- )
9 0 ?=0 om sökord ej funnet, ?=1 om ord funnet, n=pos där funnet
10 0= IF DROP CR .PAD ." EJ FUNNET I SCR #" J 3 .R 0
11 ELSE J BLEDScreen ! 0 ladda var med blocknr fr BLEDDRA
12 J WHERE 0 visa med EDITOR block o pos
13 -1 THEN ; 0 placera flagga
14
15

```

SCR #36

```

0 ( Sök ord i screen, 3 av 4 )
1 : SEARCH ( a1 a2--n3 ?)
2 0 a1=adr t sökord, a2 t basstr(med count) ?=0 sökord ej funnet
3 0 ?=1 sökordet funnet, n3=antal tkn fr start där funnet
4 SEARCH-PREP 0 dela a1 och a2 mm
5 0 DO DROP 0 starta loop drop flagga
6 3DUP 0 duplicera för JMF$ 3 stack-element
7 JMF$ DUP 0= 0 jämför o dup flagga, vänd flagga
8 IF LEAVE 0 om strängarna olika, lämna nästa LOOP
9 THEN 1 UNDER+ 0 öka adressen till basstr med 1
10 LOOP 0=>R 0 loop slut. vänd flaggan o lägg undan den
11 LET.BAS $ ->R 0 beräkna var i blocket ordet hittats
12 DROP DROP 0 drop a1 och n
13 R> R> ; 0 lägg pos o flagga på stacken
14
15

```

SCR #37

```

0 ( Sök ord i screen, 4 av 4 LOCATE )
1 : FINN ( -- ) ( FINN ord ) 0 lämnar ordet i PAD om funnet
2 -FIND IF SWAP DROP 0 lämna ordlängd på stacken
3 HERE PAD ROT CMOVE 0 flytta ordet till PAD
4 ELSE CR ." EJ FUNNET I ORDLISTAN" THEN ;
5 : LETA PAD LET.BAS $ SEARCH SEARPR ;
6 : BLEDDRA
7 DO ?TERMINAL IF LEAVE 0 avbryt bläddring med FCTN 4
8 THEN 1 BLOCK LET.BAS ! LETA 0 läs in block och leta
9 IF LEAVE ELSE EMPTY-BUFFERS 0 förhindra FLUSH
10 THEN LOOP ;
11 : LOCATE ( n1-- ) ( LOCATE ord ) 0 n1=startscreen för BLEDDRA
12 BLEDScreen ! FINN 90 BLEDScreen $ BLEDDRA ;
13 : P 0 fortsatt bläddra i nästa block
14 90 BLEDScreen $ 1+ BLEDDRA ;
15

```

TI FORTH --- a fig-FORTH extension

UTMANINGEN (FORTH-skärmar för LOCATE)

De här oförklarade hjälporden FH, THRU, 0, UNDER+ och 3DUP finns beskrivna i det som kallas omväxlande PROGRAMBITEN 84-5 och Årsmötesnumret 84.

Oj då!

I Programbiten 84-3 finns på sidan 29 ett litet program som är avsett att ta fram en diskettkatalog. Det står där litet ensamt och okommenterat. Det kanske skäms, ty det fungerar inte, påstår Bo-Arne Östberg. I varje fall inte på hans maskin. Där fungerar det däremot med följande ändringar:

```
Rad 515 J=-982 :: CALL
LOAD(J-6,LFN(NA$)+5,"",J-9,LEN(NA$)+8)
```

```
Rad 540 IF I>LEN(NA$) THEN X=0 :: GOTO
560
```

Små ändringar i programmet kan, säger Bo-Arne, medföra stor ändring av värdet på J, dvs den adress i expansionsminnet där A i AXXXXXXXXX hamnar.

Tack, Bo-Arne!

Oj-øj då!

Den alerte Jan Alexandersson har också tittat i nr 84-3 och hittat saker:

På sidan 29 i PROGRAMBITEN 84-3 fanns ett program för att köra direkt efter listningen av DISK KATALOG. Detta fungerar ej för mig. Man kan testa följande i kommando- mod utan radnummer:

```
FOR J=-1000 TO 0 :: CALL PEEK(J,A) ::
PRINT J;A;CHR$(A) :: NEXT J
```

Jag får då J=-804. Följande måste ändras:

```
515 J=-804
517 CALL LOAD(J-6,LEN(NA$)+5)
540 IF I>LEN(A$) THEN X=0 :: GOTO 560
```

Om man endast skriver rad 590 så får man J=-35. Denna position bibehålles även om flera rader med lägre nummer skrives till. Om programmet sparas med SAVE och plockas tillbaka med OLD kommer positionen att ändras.

Man kan även prova med RUN "DSK1.TEST" i kommandomod vilket ej fungerar. Minnesadress J=-810 visar längden på programnamnet. Eftersom DSK1. består av 5 tecken måste denna adress ändras till 5+NA\$.

MATERIAL TILL TIDNINGEN

Synpunkter från redaktören

Jag vill gärna ha material till tidningen. Mycket material och material av alla de sorter - program, tips om tricks, förklaringar till hur maskin och publicerade och generella program fungerar (eller inte fungerar), tekniska beskrivningar, berättelser om datorns användning, tyckar-brev om ditt och datt, rescensioner (av böcker, artiklar, andra tidningar inrikes och utrikes, program) - kort sagt om allt möjligt som kan intressera läsarna.

Som jag framhållit tidigare är det speciellt önskvärt att få material om basmaskinen - 16k, Basic/X-Basic, kassetband - och material som passar de användare som ej är så avancerade ännu.

Ju mer material som kommer, desto mer får vi att ta hand om. Kärt besvär, men ju mera av det rena slitgörat som kan rationaliseras, desto bättre. Det betyder att jag vill ha materialet i god ordning och på kassett eller på diskett. Det kan vara inskrivet med TI-Writer, Companion, Ed/Ass-editorn eller P:TEXTIN. LT-Writer, som publiceras i det här numret, blir nog också ett alternativ.

Jag vill ha en lapp med kassetten/disketten som berättar vad som finns på den och hur jag bör förfara för att komma åt innehållet. Alla längre resonemang och förklaringar, antingen de är avsedda för mig eller för publicering, kan gärna ligga på kassetten/disketten. Det är bra om texten är utskriven på printer, men jag måste ändå göra en egen som passar i tidningen.

Hänvisningar till äldre tidningsnummer och andra källor vill jag ha tydliga och riktiga. Gissningslekar kan vara roliga, men inte i det här sammanhanget.

Program som skall listas och publiceras vill jag ha som källkod, som jag kan lista upp direkt ur maskinen. Det är inte alltid jag eller någon medarbetare hinna testa programmet. Jag vill kunna lita på att Du gjort det. Skulle Du vara medveten om kvarstående 'bugs' vill jag att Du talar om detta så att vi kan påpeka det för läsarna om det ändå skulle vara intressant att publicera programmet.

Skriv tydliga körinstruktioner för program! Tala om språk, maskinkrav och varianter/valmöjligheter som finns. Berätta hur man kommer igång och vilka möjligheter man har att trassla till det för sig.

Beskriv algoritmen/teorin/tanken bakom programmet så enkelt och tydligt som möjligt. Beskriv den använda programmeringstekniken och programmets uppbyggnad så att andra kan läsa och följa programmet i programlistan.

Dokumentera i programlistan så mycket som möjligt. Knappa minimum in programmets namn, vem som skrivit programmet, datum och några rader om vad programmet är bra för.

Se till att Basic/X-Basic-programmen är omnummerade (RESQUENCE) innan Du kommenterar dem och skickar iväg dem.

Till Dig som skriver sällan eller tycker det är svårt vill jag säga: Skriv som till någon speciell person Du berättar för! Förklara tydligt vad Du vill säga. Använd gärna korta men fullständiga meningar. Använd enkla ord och absolut inga ord vars betydelse Du är oklar över. Att Du skriver till en läsare betyder också att Du bör använda sådana tilltalsord. Skriv alltså "Du" eller tala om "läsarna" - skriv inte "ni". Den som läser Din text är ju i allmänhet ensam när han/hon gör det. Skulle ändå det Du åstadkommer inte bli bra i Dina ögon - skicka det i alla fall så kan vi få igång ett resonemang! Jag rättar och härjar en hel del i det som kommer in - mest stavfel och meningsbyggnadsfel, rubriksättning, indelning i stycken mm sådant. Men det är klart - jag vill ju inte skriva Din artikel åt Dig. Skicka mig därför inte det allra första utkastet utan jobba litet med alstret, vrid och vänd på det, fundera på logiken osv.

Till slut: korta artiklar är alltid lättare att ta in än långa. Blir det mycket långt kan Du tänka på att publicera det i flera tidningar efter varandra.

Välkommen med Dina bidrag!

SÄLJES!

Fristående RS232 interface för TI 99/4A.
Pris 750:- kr.

Radskrivare Stansaab 3572. Saknar lilla å och ö.
Pris 1 000:- kr.

Anders Virving. Tfn hem 018/ 38 52 01
arb 08/780 35 08

Oj-oj-oj då!

Mera klant! Maurice Swinnen, vår käre vän i USA som nu givit sig in i programvarubranschen som exportchef för ett företag, håller verkligen ett öga på oss! Litet elakt men roligt påpekar han att ett fel han tidigare uppmärksammat oss på i ett program (det gäller Micael Dahlquists program i 84-1, sid 26 med "rättelse" i 84-2 sid 3) blivit "rättat" så att den andra villan blivit värre än den första. Så vi tar sats igen: Stark koncentration nu på programraderna 190 och 200! På båda raderna skall det stå I1 före parentesen med SEG\$. Det gör så i "rättelsen" men inte i originalet. Sedan skall det stå I2 - inte 12 - efter A\$. Det gör så i originalet men inte i "rättelsen".

Vågar jag berätta att Maurice i det brev där han skriver om det här har gjort ett nytt fel! Men det beskriver jag faktiskt inte!

Säljes:

TI-99/4A med expansionbox,
minnesexpansion,
diskkontroller,
p-kod kort,
RS 232 interface,
termoskrivare,
joysticks

Programspråk:
Logo II, Editor/Assembler samt alla andra språk

Spel:
Adventure, Parsec, TI Invaders, Moon Mine och massor av andra. Även vissa spel på diskett.

Övrigt:
Disk Manager 2, Terminal Emulator II, med mera.

Fullständig lista skickas på begäran och mot svarsporto

Detta är systemet som Björn Gustavsson utvecklade Föreningens FORTH med, troligen det mest fullständiga TI-99/4A-systemet i Sverige.

Minimipris är 13 000 SEK eller högstbjudande.

Örjan Gustavsson 0240-74172
Nordlandervägen 12 A
777 00 Smedjebacken

Koefficienter för andragradsekvation

LENNART FELLÄNDER

Program för TI-59.

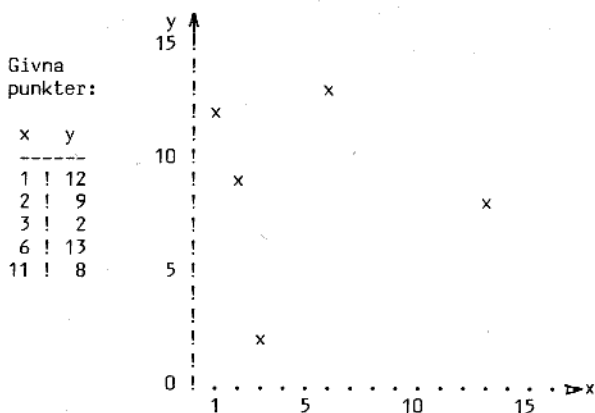
Härmed två program som kanske kan vara av intresse. Programmen hör egentligen ihop men kan även användas var för sig. De fordrar skrivare och berör båda den allmänna formen för en ekvation av andra graden med två obekanta, nämligen:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

Program nr 1 bestämmer koefficienterna A - F för den kurva som går genom fem bestämda punkter med givna koordinater.

Program nr 2 undersöker vad för slags andra-gradskurva eller kägelsnitt det är frågan om med dessa koefficienter, samt ger även möjlighet att plotta kurvan.

I program nr 1 sker beräkningen av koefficienterna med determinanter som omhändertas av biblioteksprogram 02 i standardmodulen (MASTER 1). Nedanstående exempel visar hur determinanterna uppställs:



Man utgår från determinanten för koefficienten F, som blir:

	x^2	xy	y^2	x	y
F =	1	12	144	1	12
	4	18	81	2	9
	9	6	4	3	2
	36	78	169	6	13
	121	88	64	11	8

De övriga koefficienterna erhålles genom att i ordningsföljd byta ut kolumnerna mot -1 enligt:

A =	-1	12	144	1	12
	-1	18	81	2	9
	-1	6	4	3	2
	-1	78	169	6	13
	-1	88	64	11	8

B =	1	-1	144	1	12
	4	-1	81	2	9
	9	-1	4	3	2
	36	-1	169	6	13
	121	-1	64	11	8

C =	1	12	-1	1	12
	4	18	-1	2	9
	9	6	-1	3	2
	36	78	-1	6	13
	121	88	-1	11	8

D =	1	12	144	-1	12
	4	18	81	-1	9
	9	6	4	-1	2
	36	78	169	-1	13
	121	88	64	-1	8

E =	1	12	144	1	-1
	4	18	81	2	-1
	9	6	4	3	-1
	36	78	169	6	-1
	121	88	64	11	-1

Med indelning 8 2nd OP 17 inknappas programmet enligt listning samt inspelas på block 1 och 2 på magnetkort. För att använda programmet lägger man först i x_1 -koordinaten i displayen och trycker in A-tangenten. Därpå lägger man in x_2 och trycker in B o.s.v. För y-koordinaterna trycker man på motsvarande sätt in 2nd A, 2nd B o.s.v. För exemplet ovan får man sålunda

$x_1 = 1$ och trycker A
 $x_2 = 2$ " " B
 $x_3 = 3$ " " C
 $x_4 = 6$ " " D
 $x_5 = 11$ " " E
 $y_1 = 12$ och trycker 2nd A = A'
 $y_2 = 9$ " " " B = B'
 $y_3 = 2$ " " " C = C'
 $y_4 = 13$ " " " D = D'
 $y_5 = 8$ " " " E = E',

varvid programmet startar.

Programmet beräknar automatiskt kolumnerna i determinanterna, beräknar determinanterna = koefficienterna samt byter i ordning ut kolumnerna mot -1 för respektive determinant. Programmet, som är direktadresserat, använder register 00 - 73, varvid registren 01 - 37 reserveras av biblioteksprogram 02. Detta program envisas med att före varje determinant trycka kolumnerna i en rad i skrivaren, något som är utan intresse annat än för en eventuell kontroll, och bara förbrukar papper, men som inte går att undvika. Beräkningen av determinanterna går relativt långsamt och hela programmet tar något över 9 minuter att köra.

Exemplet ovan ger:

A = 48 640
 B = 65 600
 C = 3 520
 D = -1 167 360
 E = -283 200
 F = 3 223 040

Givetvis går det att förkorta koefficienterna, något som programmet inte utför.

Exemplet i program 1

5.	5.	5.	5.	5.	5.
-1.	1.	1.	1.	1.	1.
-1.	4.	4.	4.	4.	4.
-1.	9.	9.	9.	9.	9.
-1.	36.	36.	36.	36.	36.
-1.	121.	121.	121.	121.	121.
12.	-1.	12.	12.	12.	12.
18.	-1.	18.	18.	18.	18.
6.	-1.	6.	6.	6.	6.
78.	-1.	78.	78.	78.	78.
88.	-1.	88.	88.	88.	88.
144.	144.	-1.	144.	144.	144.
81.	81.	-1.	81.	81.	81.
4.	4.	-1.	4.	4.	4.
169.	169.	-1.	169.	169.	169.
64.	64.	-1.	64.	64.	64.
1.	1.	1.	-1.	1.	1.
2.	2.	2.	-1.	2.	2.
3.	3.	3.	-1.	3.	3.
6.	6.	6.	-1.	6.	6.
11.	11.	11.	-1.	11.	11.
12.	12.	12.	12.	-1.	12.
9.	9.	9.	9.	-1.	9.
2.	2.	2.	2.	-1.	2.
13.	13.	13.	13.	-1.	13.
8.	8.	8.	8.	-1.	8.
48640.	65600.	3520.	-1167360.	-283200.	3223040.
48640. A	65600. B	3520. C	-1167360. D	-283200. E	3223040. F

Program 1

Koefficienter för andragradsekvation
8 2:nd OP 17

000	76	LBL	043	33	X ²	086	18	C'	129	42	STD	172	02	02	215	71	71
001	11	A	044	42	STD	087	49	PRD	130	41	41	173	13	C	216	04	4
002	47	CNS	045	66	66	088	45	45	131	42	STD	174	69	DP	217	03	3
003	42	STD	046	43	RCL	089	42	STD	132	42	42	175	06	06	218	42	STD
004	43	43	047	56	56	090	60	60	133	01	1	176	69	DP	219	72	72
005	42	STD	048	91	R/S	091	33	X ²	134	03	3	177	00	00	220	06	6
006	53	53	049	76	LBL	092	42	STD	135	42	STD	178	36	PGM	221	03	3
007	33	X ²	050	15	E	093	50	50	136	69	69	179	00	00	222	42	STD
008	42	STD	051	42	STD	094	43	RCL	137	03	3	180	98	ADV	223	73	73
009	63	63	052	47	47	095	60	60	138	08	8	181	29	CP	224	43	RCL
010	43	RCL	053	42	STD	096	91	R/S	139	42	STD	182	43	RCL	225	68	68
011	53	53	054	57	57	097	76	LBL	140	70	70	183	69	69	226	44	SUM
012	91	R/S	055	33	X ²	098	19	D'	141	43	RCL	184	75	-	227	71	71
013	76	LBL	056	42	STD	099	49	PRD	142	69	69	185	02	2	228	44	SUM
014	12	B	057	67	67	100	46	46	143	69	DP	186	01	1	229	72	72
015	42	STD	058	43	RCL	101	42	STD	144	04	04	187	95	=	230	73	RC*
016	44	44	059	57	57	102	61	61	145	05	5	188	67	EQ	231	73	73
017	42	STD	060	91	R/S	103	33	X ²	146	36	PGM	189	00	00	232	63	EX*
018	54	54	061	76	LBL	104	42	STD	147	02	02	190	12	12	233	71	71
019	33	X ²	062	16	A'	105	51	51	148	11	A	191	43	RCL	234	63	EX*
020	42	STD	063	49	PRD	106	43	RCL	149	01	1	192	69	69	235	72	72
021	64	64	064	43	43	107	61	61	150	36	PGM	193	75	-	236	72	ST*
022	43	RCL	065	42	STD	108	91	R/S	151	02	02	194	01	1	237	73	73
023	54	54	066	58	58	109	76	LBL	152	12	B	195	07	7	238	01	1
024	91	R/S	067	33	X ²	110	10	E'	153	73	RC*	196	95	=	239	44	SUM
025	76	LBL	068	42	STD	111	49	PRD	154	70	70	197	67	EQ	240	71	71
026	13	C	069	48	48	112	47	47	155	36	PGM	198	02	02	241	44	SUM
027	42	STD	070	43	RCL	113	42	STD	156	02	02	199	04	04	242	72	72
028	45	45	071	58	58	114	62	62	157	91	R/S	200	01	1	243	44	SUM
029	42	STD	072	91	R/S	115	33	X ²	158	01	1	201	61	GTD	244	73	73
030	55	55	073	76	LBL	116	42	STD	159	44	SUM	202	02	02	245	97	D82
031	33	X ²	074	17	B'	117	52	52	160	70	70	203	05	05	246	00	00
032	42	STD	075	49	PRD	118	43	RCL	161	43	RCL	204	04	4	247	02	02
033	65	65	076	44	44	119	62	62	162	70	70	205	44	SUM	248	30	30
034	43	RCL	077	42	STD	120	66	PAU	163	75	-	206	69	69	249	05	5
035	55	55	078	59	59	121	01	1	164	06	6	207	05	5	250	44	SUM
036	91	R/S	079	33	X ²	122	94	+/-	165	02	2	208	42	STD	251	68	68
037	76	LBL	080	42	STD	123	42	STD	166	95	=	209	00	00	252	61	GTD
038	14	D	081	49	49	124	38	38	167	22	INV	210	03	3	253	01	01
039	42	STD	082	43	RCL	125	42	STD	168	67	EQ	211	08	8	254	41	41
040	46	46	083	59	59	126	39	39	169	01	01	212	42	STD			
041	42	STD	084	91	R/S	127	42	STD	170	53	53	213	70	70			
042	56	56	085	76	LBL	128	40	40	171	36	PGM	214	42	STD			

Vilket kägelsnitt anger andragradsekvationen

Program nr 2, som undersöker vilket kägelsnitt koefficienterna A - F representerar, är baserat på en utredning i Hütte I, uppl. 26, sid. 116, där nedanstående uppställning återfinnes:

! Medelpunktskurvor $\Delta \neq 0$	!	$\Delta = 0$
! $\Delta > 0$	!	$\Delta < 0$
! Ellipsfall	!	Paraboliska fall
!	!	Hyperbelfall

! $J \neq 0$! Ja_1 el. $Ja_2 > 0$!	!	
! Äkta ! Ellips	!	Hyperbel
! kägel-! Ja_1 och $Ja_2 < 0$!	!	Parabel
! snitt ! Imag. ellips	!	

! $J = 0$! Två linjer med ändlig	!	Två parallella linjer
! Öäkt ! skärningspunkt	!	b_1 eller $b_2 =$
! kägel-!	!	> 0 ! $= 0$! < 0 !
! snitt !	!	Reella o. ! Reella o. !
! ! Imaginära ! Reella	!	skilda ! sammanf ! Imagin !
!	!	!

$$\Delta = AC - B^2/4; \quad b_1 = D^2/4 - AF; \quad b_2 = E^2/4 - CF;$$

$$J = \frac{E(AE - BD) + CD^2}{4} - \Delta F; \quad S = \frac{A + C}{2};$$

$$W = \sqrt{S^2 - \Delta}; \quad a_1 = S - W; \quad a_2 = S + W;$$

Om $A = C$; $B = 0$; $Ja_1 > 0$; föreligger en cirkel.

Programmet beräknar medelpunktens koordinater om kurvan är en ellips, cirkel eller hyperbel:

$$x_m = \frac{BE - 2CD}{4AC - B^2}; \quad y_m = \frac{BD - 2AE}{4AC - B^2};$$

För en eventuell ellips beräknas 1/2 storaxeln A och 1/2 lillaxeln B, samt för en cirkel radien R:

$$R = A = \sqrt{\frac{-F^1}{S_1}}; \quad B = \sqrt{\frac{-F^1}{S_2}};$$

$$F^1 = Dx_m + Ey_m + 2F;$$

$$S_{1,2} = A + C \pm \sqrt{(A - C)^2 + B^2};$$

För ellips, hyperbel och (parabel) beräknar programmet vinkeln $FI^0 = (\varphi^0)$ mot x-axeln. Då man samtidigt kan byta tecken för alla koefficienterna i andragradsekvationen, beror tecknet för FI^0 på tecknet för B, varför man av kurvans utseende får avgöra vilket tecken för FI^0 som skall gälla:

$$FI^0 = \frac{\text{Arcsin} \frac{-B}{\sqrt{(A + C)^2 - (4AC - B^2)}}}{2}$$

För en eventuell parabel erhålles dock rätt tecken, då vinkeln mot x-axeln beräknas ur:

$$FI^0 = \text{Arctan} - \sqrt{\frac{A}{C}};$$

Med indelning 2 Op 17 inknappas programmet enligt listning samt inspelas om så önskas på två kort på block 1, 2, 3 och 4.

För att använda programmet, lägger man i ordning in koefficienterna i displayen samt trycker på label-tangenterna enligt nedan:

Koefficient A, tryck A
" B, " B
" C, " C
" D, " D
" E, " E
" F, " E',

varvid programmet startar.

Obs! Samtliga koefficienter måste läggas in i ordning, även om de skulle råka vara noll!

Med koefficienterna beräknade ur exemplet för program 1, erhålles följande svar:

HYPERBEL
2.862... XM
13.549... YM
(-)27.739... FI

Man kan även använda programmet för att plotta fram kurvan punkt för punkt. Härvid beräknas y-koordinaterna för insatta x-värden. Om x-värdet skulle råka ligga utanför kurvan blinkar displayen och ? uppenbarar sig i skrivaren. Man trycker CLR och försöker med ett nytt x-värde.

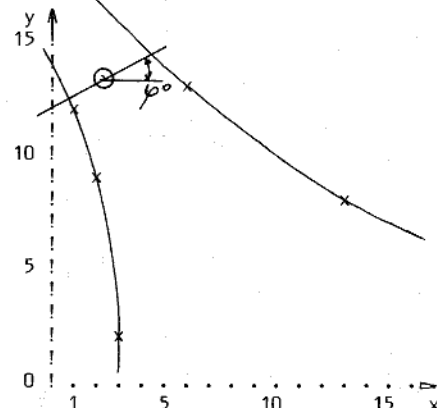
Programmet beräknar:

$$y_{1,2} = -\frac{Bx + E}{2C} \pm \sqrt{\frac{(Bx + E)^2}{4C^2} - \frac{x(Ax + D) + F}{C}};$$

Man lägger in det aktuella x-värdet i displayen och trycker 2nd A'. Kurvan med koefficienterna ur exemplet för program 1, visar sig efter plottning bli som i fig. nedan:

Plottade punkter:

x	y
1	12
2	9
3	2
6	13
11	8



Vilket kägelsnitt anger andragradsekvationen

2 2:nd OP 17

000	76	LBL	100	77	GE	200	71	SBR	300	75	-	400	69	DP	500	69	DP	600	05	5	700	00	0
001	11	A	101	04	04	201	01	01	301	43	RCL	401	01	01	501	04	04	601	01	1	701	69	DP
002	42	STD	102	20	20	202	70	70	302	18	18	402	02	2	502	43	RCL	602	03	3	702	01	01
003	00	00	103	71	SBR	203	92	RTN	303	95	=	403	07	7	503	14	14	603	01	1	703	01	1
004	91	R/S	104	02	02	204	43	RCL	304	22	INV	404	02	2	504	69	DP	604	04	4	704	07	7
005	76	LBL	105	04	04	205	01	01	305	49	PRD	405	07	7	505	06	06	605	01	1	705	03	3
006	12	B	106	71	SBR	206	65	X	306	14	14	406	01	1	506	69	DP	606	07	7	706	01	1
007	42	STD	107	03	03	207	43	RCL	307	85	+	407	03	3	507	00	00	607	02	2	707	03	3
008	01	01	108	27	27	208	04	04	308	02	2	408	00	0	508	98	ADV	608	07	7	708	06	6
009	91	R/S	109	02	2	209	75	-	309	65	X	409	00	0	509	91	R/S	609	71	SBR	709	04	4
010	76	LBL	110	03	3	210	02	2	310	43	RCL	410	02	2	510	43	RCL	610	01	01	710	00	0
011	13	C	111	04	4	211	65	X	311	18	18	411	07	7	511	08	08	611	46	46	711	02	2
012	42	STD	112	05	5	212	43	RCL	312	95	=	412	71	SBR	512	77	GE	612	71	SBR	712	07	7
013	02	02	113	03	3	213	02	02	313	22	INV	413	01	01	513	05	05	613	03	03	713	71	SBR
014	91	R/S	114	03	3	214	65	X	314	49	PRD	414	70	70	514	38	38	614	74	74	714	01	01
015	76	LBL	115	69	DP	215	43	RCL	315	15	15	415	91	R/S	515	43	RCL	615	98	ADV	715	70	70
016	14	D	116	02	02	216	03	03	316	43	RCL	416	71	SBR	516	09	09	616	91	R/S	716	91	R/S
017	42	STD	117	01	1	217	95	=	317	14	14	417	01	01	517	77	GE	617	43	RCL	717	76	LBL
018	03	03	118	07	7	218	55	+	318	34	FX	418	86	86	518	05	05	618	03	03	718	16	A
019	91	R/S	119	03	3	219	53	(	319	42	STD	419	91	R/S	519	38	38	619	33	X²	719	42	STD
020	76	LBL	120	05	5	220	04	4	320	14	14	420	43	RCL	520	02	2	620	55	+	720	07	07
021	15	E	121	01	1	221	65	X	321	43	RCL	421	00	00	521	04	4	621	04	4	721	65	X
022	42	STD	122	04	4	222	43	RCL	322	15	15	422	85	+	522	69	DP	622	75	-	722	43	RCL
023	04	04	123	01	1	223	00	00	323	34	FX	423	43	RCL	523	01	01	623	43	RCL	723	01	01
024	91	R/S	124	07	7	224	65	X	324	42	STD	424	02	02	524	03	3	624	00	00	724	85	+
025	76	LBL	125	02	2	225	43	RCL	325	15	15	425	95	=	525	00	0	625	65	X	725	43	RCL
026	10	E	126	07	7	226	02	02	326	92	RTN	426	55	+	526	01	1	626	43	RCL	726	04	04
027	42	STD	127	71	SBR	227	75	-	327	43	RCL	427	02	2	527	03	3	627	05	05	727	95	=
028	05	05	128	01	01	228	43	RCL	328	00	00	428	95	=	528	02	2	628	95	=	728	55	+
029	43	RCL	129	46	46	229	01	01	329	85	+	429	42	STD	529	02	2	629	42	STD	729	02	2
030	00	00	130	71	SBR	230	33	X²	330	43	RCL	430	08	08	530	04	4	630	10	10	730	55	+
031	65	X	131	03	03	231	54	)	331	02	02	431	42	STD	531	00	0	631	43	RCL	731	43	RCL
032	43	RCL	132	53	53	232	42	STD	332	95	=	432	09	09	532	01	1	632	04	04	732	02	02
033	04	04	133	71	SBR	233	17	17	333	33	X²	433	33	X²	533	07	7	633	33	X²	733	95	=
034	75	-	134	03	03	234	95	=	334	75	-	434	75	-	534	71	SBR	634	55	+	734	94	+/-
035	43	RCL	135	74	74	235	42	STD	335	43	RCL	435	43	RCL	535	01	01	635	04	4	735	42	STD
036	01	01	136	98	ADV	236	12	12	336	17	17	436	06	06	536	54	54	636	75	-	736	08	08
037	65	X	137	91	R/S	237	43	RCL	337	95	=	437	95	=	537	91	R/S	637	43	RCL	737	42	STD
038	43	RCL	138	58	FIX	238	01	01	338	34	FX	438	34	FX	538	71	SBR	638	02	02	738	09	09
039	03	03	139	08	08	239	65	X	339	35	1/X	439	44	SUM	539	02	02	639	65	X	739	33	X²
040	95	=	140	52	EE	240	43	RCL	340	65	X	440	09	09	540	04	04	640	43	RCL	740	75	-
041	65	X	141	22	INV	241	03	03	341	43	RCL	441	22	INV	541	71	SBR	641	05	05	741	53	(
042	43	RCL	142	52	EE	242	75	-	342	01	01	442	44	SUM	542	02	02	642	95	=	742	43	RCL
043	04	04	143	22	INV	243	02	2	343	94	+/-	443	08	08	543	58	58	643	42	STD	743	07	07
044	85	+	144	58	FIX	244	65	X	344	95	=	444	43	RCL	544	71	SBR	644	11	11	744	65	X
045	43	RCL	145	92	RTN	245	43	RCL	345	22	INV	445	07	07	545	03	03	645	77	GE	745	53	(
046	02	02	146	69	DP	246	00	00	346	38	STN	446	49	PRD	546	27	27	646	06	06	746	43	RCL
047	65	X	147	03	03	247	65	X	347	55	+	447	08	08	547	01	1	647	57	57	747	00	00
048	43	RCL	148	69	DP	248	43	RCL	348	02	2	448	49	PRD	548	07	7	648	43	RCL	748	65	X
049	03	03	149	05	05	249	04	04	349	95	=	449	09	09	549	71	SBR	649	10	10	749	43	RCL
050	33	X²	150	69	DP	250	95	=	350	42	STD	450	43	RCL	550	01	01	650	77	GE	750	07	07
051	95	=	151	00	00	251	55	+	351	16	16	451	00	00	551	54	54	651	06	06	751	85	+
052	55	+	152	98	ADV	252	43	RCL	352	92	RTN	452	75	-	552	71	SBR	652	57	57	752	43	RCL
053	04	4	153	92	RTN	253	17	17	353	04	4	453	43	RCL	553	03	03	653	71	SBR	753	03	03
054	95	=	154	69	DP	254	95	=	354	04	4	454	02	02	554	53	53	654	01	01	754	54	)
055	42	STD	155	02	02	255	42	STD	355	03	3	455	95	=	555	01	1	655	86	86	755	85	+
056	07	07	156	02	2	256	13	13	356	00	0	456	22	INV	556	03	3	656	91	R/S	756	43	RCL
057	43	RCL	157	07	7	257	92	RTN	357	69	DP	457	67	EQ	557	69	DP	657	43	RCL	757	05	05
058	00	00	158	02	2	258	43	RCL	358	04	04	458	05	05	558	04	04	658	10	10	758	54	)
059	65	X	159	07	7	259	03	03	359	43	RCL	459	10	10	559	43	RCL	659	71	SBR	759	55	+
060	43	RCL	160	02	2	260	65	X	360	12	12	460	43	RCL	560	14	14	660	01	01	760	43	RCL
061	02	02	161	04	4	261	43	RCL	361	69	DP	461	01	01	561	69	DP	661	38	38	761	02	02
062	75	-	162	03	3	262	12	12	362	06	06	462	22	INV	562	06	06	662	67	EQ	762	95	=
063	43	RCL	163	03	3	263	85	+	363	04	4	463	67	EQ	563	01	1	663	06	06	763	34	FX
064	01	01	164	03	3	264	43	RCL	364	05	5	464	05	05	564	04	4	664	95	95	764	44	SUM
065	33	X²	165	06	6	265	04	04	365	03	3	465	10	10	565	69	DP	665	43	RCL	765	09	09
066	55	+	166	71	SBR	266	65	X	366	00	0	466	43	RCL	566	04	04	666	11	11	766	22	INV
067	04	4	167	01	01	267	43	RCL	367	69	DP	467	08	08	567	43	RCL	667	71	SBR	767	44	SUM
068	95	=	168	46	46	268	13	13	368	04	04	468	22	INV	568	15	15	668	01	01	768	08	08
069	42	STD	169	92	RTN	269	85	+	369	43	RCL	469	77	GE	569	69	DP	669	38	38	769	04	4
070</																							

FORTH spalten

REDAKTÖR: Lars-Erik Svahn
Mellanbergsu. 23
13545 TYRESÖ
tel: 08-742 61 07

OBS! %-tecknet i den här artikeln står för FORTH-tecknet alfa-slang (=ASCII 64), FORTH-ordet "fetch". Tyvärr saknas det tecknet på den skrivare som använts. OBS!

Den här spalten ska handla om Forth, och speciellt förstås om Forth på 99-an. Jag har också blivit ombedd att se till att Forth jämförs med andra språk, och då särskilt med TI Basic. Det är tänkt att innehållet ska baseras på material insänt av Dig, kära läsare: finurliga program av "experter" som jag själv (jag har hållit på ett helt år (!) med Forth) eller lätt besvarade frågor av dem som ännu ej har uppnått den fullständiga KLARHETEN. I den här första utgåvan av spalten tänker jag jämföra de fyra vanligaste språken till 99-an, nämligen: Assembler, Basic, Forth och Pascal. Vilken utrustning som behövs för att använda de olika språken finns att läsa i Programbiten / Nittinian nr 83-2. Jag tänker bara tillägga att en minimi utrustning för Forth är: minnes expansion (eventuellt fristående, nypris c:a 1500 kr); kassett-bandspelare; Mini Memory eller Extended Basic-modulen, nypris c:a 1000 kr; föreningens Forth på kassett, 250 kr. Jag uppfattar att det finns fyra grundläggande aspekter på ett programmerings språk, nämligen enkelhet, generalitet, säkerhet och effektivitet.

Enkelhet

Med enkelhet kan man mena olika saker, t ex att språket ska vara lätt att lära sig; att språket ska ge lättlästa program; att det ska gå snabbt att programmera; att det ska vara enkelt att föra en dialog med datorn

I detta sammanhang tänker nog de flesta på Basic. Basic-programmering är ganska lätt i början, men Du slår snart huvudet i taket och får program med Spagetti-Syndromet: program som är så svåra att följa att inte ens Du själv förstår eller kan avlusa. Med Forth är det precis tvärt om: det är litet besvärligt i början, men sedan öppnas en lucka i taket och Du ser stjärnorna. När Du har läst halva "STARTING FORTH" av Leo Brodie och tränat på sidan av med datorn, så har nog Din lust att stängas med Basic minskat avsevärt. Efter 1 år med Forth och hela STARTING FORTH tycker jag själv att det är svårt att skriva Basic program. Enligt mitt tycke är det lättare att läsa Forth-program än någon annan sorts program (Basic program är ovanligt svårslästa). En oinvidd som ser en Forth-skärm med definitioner har antagligen svårt att förstå detta, men man måste komma ihåg att i Forth programmerar man i nivåer, och att de lägsta nivåerna snarast är att jämföra med assembler-program. Forth på högsta nivå kan mycket väl vara ren engelska eller svenska. Det går FORT att programmera i FORTH, när man blivit van. Om man bara tänker efter vilka "delprogram" (Forth skiljer inte på subrutiner och program - alla definitioner har samma status) som kommer att behövas och först definierar dessa, så blir man nästan alltid förvånad över hur lättåtkomlig problemets lösning är. Att programmera i Basic känns nästan som att sitta i en bilkö, med GOTO satser som hänvisar till rader man ännu inte skrivit etc.

Basic var en gång det enda språk som kunde kallas dialog språk. Utvecklingen har nått långt sedan dess. När jag försöker föra en dialog med hjälp av Basic blir det bara RUN RUN RUN hela tiden. När jag för en dialog med hjälp av Forth blir det kanske:
å 34 41 49 å ALL LOAD svar: "ok"
PR 22 LIST svar: utskrift på skrivaren och "ok"
SWEDISH CHARACTERS svar: "? SWEDISH"
70 LOAD SWEDISH CHARACTERS svar: "ok"
13 LOAD svar: " ? : definition not finished!"
WHERE svar: här hamnar jag i editorn i den skärm och på den rad där felet uppstod då jag försökte kompilera skärmen. Orv.

Det finns också en annan aspekt på enkelhet: både systemprogram och användarprogram ska vara enkla och lätta att förstå. Just denna aspekt på enkelheten är Forths grundfilosofi och Forth-programmerarens ledstjärna. Nästan alla systemprogrammen i Forth är lätta att förstå, göra ändringar och tillägg i.

Generalitet

Finns det verkligen något annat språk som kan användas i så många olika sammanhang som Forth? Det som inte finns definierat i Forth, definierar man själv, nästan precis som man vill ha det.

Jag har en liten fyrfärgs plotter (MCP-40 c:a 2000 kr). Till den har jag gjort ett enkelt operativsystem: 10 -30 MOVE för pennan till (10,-30) utan att rita; -101 34 DRAW får pennan att dra en rät linje från nuvarande position till (-101,34); RED PEN skiftar till röda pennan etc. Men det gör detsamma vad man vill göra - Forth går att utvidga så att det blir enkelt:
FILE: DSK1.TEST å REL INP ÅMODE 80 OPEN öppnar en fil;
MAXLOAD: CS1 MAXSAVE: DSK1.PIRATE överför ett "hemligt" äventyrsprogram från kassett till flexskiva;
10 50 SAY-LINE uttalar prat-strängen på skärm 50 rad 10.

Om man inte gillar IF ELSE ENDIF satserna i Forth (men man vänjer sig faktiskt) så kan man ändra syntaxen:

```
: THEN ÅCOMPILEA IF ; IMMEDIATE  
: IF ; IMMEDIATE
```

Nu kan IF THEN ELSE ENDIF användas ungefär som i Extended Basic och Pascal.

När jag hade tröttnat på att sitta och vänta på att den röda lampan på diskenheten skulle slockna, definierade jag ordet DISSABLE som stänger av enheten när det väsentliga är gjort. Med det motsatta ordet ACTIVATE hålls spindeln snurrande i väntan på att den långsamma skrivaren ska bli redo för nästa post.

Ett annat exempel är Björn Gustavssons suveräna skrivar-operativ-system. Man skriver bara PR framför de program, där man vill att utskrift ska ske på skrivaren istället för på skärmen:

```
PR 22 LIST eller  
PR MÅNADSRAPPORT SLUTREDOVISNING.
```

Här följer några definitioner som jag tycker är bra vid dialog med datorn:

```
: å !CSP ;  
: å SP% CSP % SWAP - 2/ ;  
: ALL ?EXEC ÅCOMPILEA " CFA >R 0  
DO J EXECUTE LOOP R> DROP ;  
: STEP ?EXEC ÅCOMPILEA " CFA >R 1+ SWAP  
DO I J EXECUTE LOOP R> DROP ;
```

Ordet å lagrar antalet tal på stacken i variabeln CSP och ordet å räknar ut hur många tal som kommit till sedan å och lagrar detta antal överst på stacken.

Exempel: å 2 3 5 7 11 13 17 å ger
stacken: 2 3 5 7 11 13 17 7.

Ordet ALL kan användas med ord som EMIT . LOAD LIST mfl, vilka har stackdiagrammet (n--), på följande sätt:

```
å 22 20 17 10 6 å ALL LOAD medför att först skärm 6  
laddas sedan skärm 10 osv tom skärm 22.
```

Ordet STEP kan användas med samma sorts ord som ALL: 20 26 STEP LIST listar skärm 20 tom skärm 26. Varken ALL eller STEP kan kompileras, utan endast kommenderas direkt (i annat fall blir det ett felmeddelande).

I detta sammanhang bör kanske Pascals datastrukturer omnämnas. Speciellt begreppet RECORD är av en generell typ: ett sorts "array" där de olika fälten kan vara av olika typ, från heltal till mängder eller tom av typ record.

Det är kanske något att implementera i Forth?

Säkerhet

Med säkerhet menas att systemet dels ska hjälpa programmeraren att hitta fel, dels förhindra programmeraren att göra svårupptäckta fel och att åstadkomma systemkrasch.

Pascal är säkert. Det är konstruerat så att syntaxfel och felstavade identifierare upptäcks vid kompileringen och så att inga "irregulära" dataöverföringar tillåts. Pascals ide bygger delvis på att förhindra programmeraren från att göra svårupptäckta fel. Fördelen med denna säkerhet är att själva programmeringen blir effektivare. En av nackdelarna är att många saker blir omöjliga att göra, eller förbjudna att göra, eftersom de är svåra för systemet att kontrollera. Detta gör Pascal till ett mindre generellt språk än ved som annars skulle vara fallet. Basic är säkert så till vida att det är svårt att åstadkomma systemkrascher (utan hjälp av CALL LOAD etc). Men om en variabel är felstavad, uppfattar systemet detta som en ny variabel (vilken ges värdet 0), ett fel vilket ibland kan vara svårt att upptäcka. Syntaxfel utpekas med radnummer och kan lätt rättas, men att vara hänvisad till GOTO för att styra programflödet ger låg säkerhet vad beträffar logiken i programmet.

I Extended Basic kan felstavnings kontroll införas med hjälp av instruktionerna !%P+ och !%P- och program flödet kan styras mer strukturerat och lättöverskådligt.

I Forth kontrolleras att alla definitioner är rätt stavade, viss syntax kontroll sker vid kompileringen och vid testkörning får man viss kontroll genom att kontrollera stacken före och efter. Det går också att göra hjälpprogram som spårar fel. Den viktigaste säkerhets åtgärden är ändå FLUSH - ett ord som överför alla gjorda ändringar i källkoden på FORTH-skärmarna till skivminnet.

Systemkrascher är svåra att helt förebygga i Forth, det måste erkännas. Därför är det viktigt att ha ett uppdaterat skivminne.

Men Forth är ju utbyggbart och kontroller går att konstruera. För att visa ett sätt att göra det kan vi ta exemplet RED PEN. Det finns i grundversionen ingen syntax kontroll för detta kommando. Systemet kanske inte upptäcker om jag skriver PEN RED. Följande omdefiniering klarar detta:

```
HEX
: RED STATE % IF COMPILE RED CO
      ELSE RED ENDIF ; IMMEDIATE
: PEN STATE % IF CO ?PAIR COMPILE PEN
      ELSE PEN ENDIF ; IMMEDIATE
```

STATE är en system variabel som EJ är lika med 0 vid kompilering och 0 annars. IMMEDIATE ser till att senast gjorda definition blir sådan att den inte kompileras när den finns med i en definition, utan att den istället exekveras under kompileringen av denna definition.

?PAIR jämför de två översta talen på stacken. Är de inte lika blir det ett felmeddelande.

Effektivitet

Under den här rubriken är det dags att nämna Assembler språket. Assembler är snabbast och kan utnyttja minnet effektivt. Men det är bara exekveringen som är snabb. Programmering, assemblering etc är ganska långsamma processer. Att programmera assembler är roligt när man vill göra allt själv från grunden. Det är nödvändigt i en del andra fall. Oftast är det trots allt lämpligare med ett högnivåspråk.

Även Forth är snabbt. Normalt och ungefärligt gäller följande jämförelse med assembler:

Forth tar dubbelt så lång tid;
Pascal tar sex till sju gånger så lång tid;
Basic tar femtio gånger så lång tid.

Vad beträffar minnesutnyttjande är Forth effektivast. En instruktion kräver i regel endast två byte, och den instruktionen kan vara ett mycket avancerat program. Men då är förstas inte den plats som systemprogrammet upptar (kärnan med vissa utökningar) inräknad. Men det gör inte så mycket, för det finns plats ändå. Och dessutom har Forth något som kallas för virtuellt minne. Detta fungerar så att man i viss mån upplever det som att skivminnet är en del av datorns "inbyggda" minne.

Forths effektivitet är en följd av dess enkla och smarta grundkonstruktion. Språket konstruerades i början av sjuttio talet av Charles Moore, en amerikansk astronom.

Det finns en grupp i USA som kallar sig Forth Interest Group (FIG). Gruppen ger ut en tidning som heter FORTH Dimensions. För oss i Europa kostar det 27 dollar att bli medlem i FIG. Med medlemskapet följer ett års prenumeration på FORTH Dimensions. Det är förresten FIG som ligger bakom den Forth-dialekt som vi har till 99-an.

Adressen till FIG är:
FORTH Interest Group
P.O. Box 1105
San Carlos, CA 94070
USA

Till sist vill jag uppmäna alla som har något intresse av Forth att skicka in material till mig. Mycket material gör en intressantare Forth-spalt. Skicka det direkt till mig under den adress som står i inledningarna till spalten.

SÄLJES!

TI 99 expansionsbox.
32K minnesexpansion, Disk Controller, Disk Drive, RS232-kort -->5 700:-

Editor/Assembler + TI Writer med dokumentation mm -->1 400:-

SEIKOSHA printer GP 100A -->1 200:-

Paketpris! ----->7 800:-

Björn Mårtensson, tfn 08/765 47 78

(Det här är vår programbankir som nu blir professionell och byter till en mera avancerad maskin. Tack och lycka till! /Lennart Lindberg)

CATALOG

av Jan Alexandersson

Det här lilla trevliga programmet använder Jan på alla sina disketter för att kunna lista innehållet utan att byta till Disk Manager-modulen. Det tar bara upp tre sektorer på skivan./Red

```
100 CALL CLEAR
110 T$="DIS/FIXDIS/VARINT/FIXINT/VARPROGRAM"
120 OPEN $1:"DSK1.",INPUT,RELATIVE,INTERNAL
130 INPUT $1:A$,J,K
140 PRINT A$;" LED=";STR$(K); ANV=";STR$(J-K):"fil
      namn sekt typ E p":
150 FOR I=1 TO 99
160 INPUT $1:A$,A,J,K
170 IF LEN(A$)=0 THEN 230
180 PRINT A$;TAB(14-LEN(STR$(J)));J;TAB(17);SEG$(T$,A
      BS(A)*7-6,7);SEG$("&STR$(K),1-(K>99),-3*(ABS(A)
      <>5));TAB(28);CHR$(32-10*(A<0))
190 CALL KEY(3,T,S)
200 IF S=0 THEN 220
210 GOSUB 260
220 NEXT I
230 CLOSE $1
240 GOSUB 260
250 END
260 CALL KEY(3,T,S)
270 IF S<1 THEN 260
280 RETURN
```

Trippelpunkten

AV ARNE WENNBORG.

på 99 an

I PROGRAMBITEN nr 84-3 finns en artikel om beräkning av trippelpunkten i en sfärisk triangel med TI-59. Med trippelpunkten menas tyngdpunkten av den buktiga ytan. T ex den punkt på jordytan som ligger lika långt från tre givna punkter.

Här följer ett program för TI-99:an i X-BASIC. Det visar att även 99:an kan räkna snabbt och med stor precision. Programmet är lätt att använda men kanske inte lika lätt att förstå. Det krävs vissa kunskaper i sfärisk trigonometri. Jag skall försöka förklara det något.

I raderna 160-190 finns definitioner av karaktärer och matematiska uttryck. Därefter i raderna 200-310 finns titelbild och instruktioner.

I raderna 320-600 mottar programmet i ACCEPT-satser de tre positionernas latitud och longitud. Lat. anges i N eller S (0-90) och long. i O eller V (0-180) båda i grader och minuter.

Önskas större noggrannhet kan minuterna slås in med två decimaler. För long. kan också E eller W användas.

I raderna 620-810 förvandlas minuter till grader och grader till radianer. Om lat. är S eller long. V bytes tecken. Om latituderna är lika ändras de genom ökning eller minskning med 0.00001 radianer (motsvarande 0.03 bågminuter). I beräkningsformlerna finns nämligen i nämnaren uttrycket (sinus för en lat. minus sinus för en annan lat.). Om lat. är lika blir nämnaren noll och bråket går mot oändligheten med resultat att NUMERIC OVERFLOW uppstår. Det kan elimineras med tillskottet.

Den västligaste orten (A) skall slås in först och den östligaste (C) sist. Om longitudsskillnaden är större än 180 grader transformeras koordinaterna av programmet och long. räknas från 180 graders meridianen med ombytt tecken.

I programmets slutskede återställs long. på nytt och räknas från Greenwich. slinga. För att öka snabbheten beräknas vissa konstanter först (830-860).

I slingan (900-1030) beräknas longituden (L) för trippelpunkten (T). Det göres genom att ständigt minskande värden av L sättes in på prov i ekvationen i rad 920.

Det rätta värdet på L är funnet när X=0. I rad 870 sättes ingångsvärden för slingan.

Slingans gränser blir: Den västligaste long. minus 90 gr. plus 90 gr. Anledningen är att T i många fall ligger utanför triangeln. Värdet för step (SP) sättes i samma rad till 10 gr (PI/18 rad) I slingan provas vid körningen till en början var tionde grad av L. Varje gång den passerar beräknas ett X-värde (i rad 860) som minskar för varje gång.

I raderna 960-1000 beräknas differensen (K) av två på varandra följande X-värde en.

När det absoluta värdet av X blir mindre än fem gånger K går programmet ur slingan till rad 1270. Där sättes L:s utgångsvärde till ingångsvärde i en ny slinga. Samtidigt halveras SP.

Programmet går tillbaka till 840 och en ny slinga körs. Förloppet upprepas åtskilliga gånger vilket leder till att X går mot noll. När X blir mindre än vad som anges i rad 1010 (1E-7) lämnar programmet slingan för gott. L-värdet är då T:s long. Genom att öka exponentielluttrycket i rad 1010 kan man snabba upp exekveringshastigheten på bekostnad av noggrannheten.

För att man skall kunna följa hur programmet arbetar är tre DISPLAY-satser inlagda i slingan.

På skärmen visas antalet beräkningar (hur många gånger slingan passerar) och den provade long. i radianer samt X-värdet. Vidare finns också värdet på SP. Det kommer dock från rad 1270 utanför slingan.

DISPLAY-satserna gör programmet långsammare och kan tas bort om så önskas. Man går då miste om nöjet att se hur X går mot noll och hur programmet snabbt kommer fram till rätt long. Det som tar tid är att finslipa resultatet.

I raderna 1050-1110 omvandlas L på nytt och används för beräkning av lat. Det senare sker i raderna 1130-1220. Beräkningarna göres två gånger med olika ingångsvärden och på de erhållna resultaten tas medelvärdet.

När lat. också förvandlats till gr. och min. skrivs lat. och long. ut på skärmen (1240-1250). Därefter beräknas för kontroll avståndet till T från de tre givna positionerna. Det göres i raderna 1300-1430 och i subrutinen i 1530. De tre avstånden skall vara lika.

Efter detta kollar programmet om den funna T är den rätta. Så är inte alltid fallet.

Vid körning kan det inträffa för vissa trianglar att programmet i slingan först kommer till ett L som skiljer sig 180 grader från det rätta. X-värdet är även då lika med noll och programmet går ur slingan. Eftersom lat. då beräknats med fel long. blir det fel också i lat. Den blir rätt numerärt men får fel namn (S istället för N och tvärtom). Är positionen på skärmen då fel? Egentligen inte. Programmet har svarat med en punkt som är antipod till den sökta T och denna är också en trippelpunkt!

Det finns en rutin som återställer ordningen. Programmet går till rad 1660. Där kollas om avstånden till T är lika och om dessa är större än 90 gr. (5400 min).

Om så är fallet har vi funnit antipoden. Då går programmet till rad 1670. Där ökas long. med 180 gr. (PI) och går tillbaka till rad 1060 för ny latituds beräkning. En ny T erhålles och värdena på skärmen ändras. Nya distanser till T räknas fram och kollas likadant.

Nu är distanserna lika och under 90 gr. De skrives oavkortade för att ge en uppfattning av precisionen. De omräknas också till kilometer och skrives på heltal. I distansformlerna beräknas också avstånden mellan de tre positionerna i distansminuter och i km.

Programmet fortsätter 1440 och GOSUB 1700 där utskrivningen av distanserna sker.

I det exempel som finns i nr 84-3 med pos. Tokyo (N 36 / O 139), Melbourne (S38 / O 145) och Los Angeles (N 35 / V 118) beräknar programmet samtliga tre orters avstånd till T till 348 1.0020 bågminuter.

Om man i rad 1010 minskar till 1E-8 får man resultaten lika på 6:e decimalen. Varje enhet i denna motsvarar 18 millimeter! Och detta på en triangel över hela Stilla Havet!

Exekveringstiden är under två minuter. Man bör dock komma ihåg att resultatet gäller för en sfär och eftersom jorden avviker från den sfäriska formen gäller inte denna noggrannhet på jordytan.

Vid beräkning av km-avstånden har använts 1852 m som medeltal för bågminutens längd. Det känns därför inte meningsfullt att redovisa km-avstånden med decimaler.

Programmet kan hantera de flesta sfäriska trianglar men det finns undantag. Om long.-differensen eller sidornas längd närmar sig 180 gr. kan polen komma att inneslutas i triangeln.

Ligger polen inne i triangeln är summan av long.-diff 360. En del sådana trianglar går inte bra men många gör det. Pröva den här! Cape of Good Hope (S34/O18), Cape Horn (S56/V67), Tasmania (S43/O147).

En sfärisk triangel kan se ut på många sätt. Vad sägs om följande? N60/V100, N60/V60, N60/V20. Alla hörnen ligger på samma latitudsparallell. T ligger då i närmaste pol.

Om två hörn lägges på ekvatorn med en long.-diff. på 90 gr. och det tredje i polen fås en triangel där samtliga tre sidor och samtliga tre vinklar är 90 gr. En sådan klarar programmet.

Det går också bra med mycket små trianglar. Vad sägs om denna över Öresund: Vedbäck hamn (N55gr 51min/012gr 34.5min), Råå hamn (N55gr 59.7min/012gr 44.8mi n), Landskrona hamn (N55gr 51.25min/012gr 50.05min). Trippelpunkten ligger 5.47715 distansminuter (ca 10 km) från de tre positionerna mitt i ruinerna av Tycho Brahes observatorium på ön Ven.

Man kan fundera över vad han kunde ha åstadkommit om han haft tillgång till en TI-99:a!

Lycka till med beräkningarna!

```

100 REM *****
110 REM *TRIPPELPUNKTEN*
120 REM *ETT PROGRAM AV*
130 REM *ARNE WENNERBERG*
140 REM *****
150 CALL CLEAR :: CALL SCREEN(11):: DIM DIS(24),KI(24)
160 REM DEFINITIONER
170 DEF ARCSIN(X)=2*ATN(X/(1+SQR(1-X*X))): DEF ARCCO
S(X)=PI/2-ARCSIN(X)
180 CALL CHAR(91,"00280038447C4444"):: CALL CHAR(92,"
0028007C4444447C"):: CALL CHAR(93,"00382838447C44
44")
190 CALL CHAR(94,"C3C7666637361E1E"):: CALL CHAR(95,"
C3E36666EC6C7878")
200 DISPLAY AT(8,1):"T R I P P E L P U N K T E N"
210 DISPLAY AT(9,1):"*****"
220 DISPLAY AT(15,12):"19° 85' :: FOR I=1 TO 800 ::
EXT I :: CALL CLEAR
230 DISPLAY AT(1,1):"I N S T R U K T I O N E R"
240 DISPLAY AT(2,1):"-----"
250 DISPLAY AT(5,1):"SLÅ IN TRE POSITIONER:"
260 DISPLAY AT(7,1):"LATITUD ANGES N ELLER S 0-";"90,
OCH LONGITUD 0 ELLER";"V 0-180, I GRADER OCH MIN
-";"UTER."
270 DISPLAY AT(12,1):"MINUTERNA KAN ANGES MED TVÅ";"D
ECIMALER OM STOR NOGGRAN";"HET ÖNSKAS."
280 DISPLAY AT(16,1):"DEN VÄSTLIGASTE POSITIONEN";"--
-----";"SLÅS IN FÖRST OCH DEN
OST-";"-----"
290 DISPLAY AT(20,1):"LIGASTE SIST." :: DISPLAY AT(21
,1):"-----"
300 DISPLAY AT(24,1):"TRYCK PÅ VALFRI TANGENT!" :: CA
LL KEY(0,KK,SS)
310 IF SS THEN 330 ELSE 300
320 REM INPUT LAT OCH LONG
330 CALL CLEAR :: CALL SCREEN(11):: DISPLAY AT(2,1):"
POS LATITUD LONGITUD"
340 FLAG=0 :: Z=0 :: N=0 :: DISPLAY AT(4,5):"N/S GR M
IN O/V GR MIN"
350 DISPLAY AT(6,2):"A" :: ACCEPT AT(6,6)SIZE(1)VALID
ATE("NS"):F1$
360 ACCEPT AT(6,9)SIZE(2)VALIDATE(DIGIT):F1G :: IF F1
G>90 THEN 360
370 IF F1G=90 THEN 390
380 ACCEPT AT(6,12)SIZE(5)VALIDATE(DIGIT,"."):F1M ::
IF F1M>60 THEN 380

```

```

390 ACCEPT AT(6,18)SIZE(1)VALIDATE("VOWE"):L1$
400 ACCEPT AT(6,20)SIZE(3):L1G :: IF L1G>180 THEN 400
410 IF L1G=180 THEN 430
420 ACCEPT AT(6,24)SIZE(5)VALIDATE(DIGIT,"."):L1M ::
IF L1M>60 THEN 420
430 DISPLAY AT(7,2):"B" :: ACCEPT AT(7,6)SIZE(1)VALID
ATE("NS"):F2$
440 ACCEPT AT(7,9)SIZE(2)VALIDATE(DIGIT):F2G :: IF F2
G>90 THEN 440
450 IF F2G=90 THEN 470
460 ACCEPT AT(7,12)SIZE(5)VALIDATE(DIGIT,"."):F2M ::
IF F2M>60 THEN 460
470 ACCEPT AT(7,18)SIZE(1)VALIDATE("OVEW"):L2$
480 ACCEPT AT(7,20)SIZE(3)VALIDATE(DIGIT):L2G :: IF L
2G>180 THEN 480
490 IF L2G=180 THEN 510
500 ACCEPT AT(7,24)SIZE(5)VALIDATE(DIGIT,"."):L2M ::
IF L2M>60 THEN 500
510 DISPLAY AT(8,2):"C" :: ACCEPT AT(8,6)SIZE(1)VALID
ATE("NS"):F3$
520 ACCEPT AT(8,9)SIZE(2)VALIDATE(DIGIT):F3G :: IF F3
G>90 THEN 520
530 IF F3G=90 THEN 550
540 ACCEPT AT(8,12)SIZE(5)VALIDATE(DIGIT,"."):F3M ::
IF F3M>60 THEN 540
550 ACCEPT AT(8,18)SIZE(1)VALIDATE("OVEW"):L3$
560 ACCEPT AT(8,20)SIZE(3)VALIDATE(DIGIT):L3G :: IF L
3G>180 THEN 560
570 IF L3G=180 THEN 590
580 ACCEPT AT(8,24)SIZE(5)VALIDATE(DIGIT,"."):L3M ::
IF L3M>60 THEN 580
590 CALL SCREEN(4):: DISPLAY AT(10,2):"TACK! RÄKNI
NG PAGARI!"
600 DISPLAY AT(16,2):"LONG (R):" :: DISPLAY AT(18,2):
"X-VÄRDE:" :: DISPLAY AT(20,2):"STEP:" :: DISPLAY
AT(12,2):"ANTAL BER.:"
610 REM TRANSFORMATIONER
620 F1=(F1G+F1M/60)/(180/PI)
630 IF F1$="S" THEN F1=-F1
640 L1=(L1G+L1M/60)/(180/PI)
650 IF L1$="V" OR L1$="W" THEN L1=-L1
660 F2=(F2G+F2M/60)/(180/PI)
670 IF F2$="S" THEN F2=-F2
680 L2=(L2G+L2M/60)/(180/PI)
690 IF L2$="V" OR L2$="W" THEN L2=-L2
700 F3=(F3G+F3M/60)/(180/PI)
710 IF F3$="S" THEN F3=-F3
720 L3=(L3G+L3M/60)/(180/PI)
730 IF L3$="V" OR L3$="W" THEN L3=-L3
740 IF ABS(L1-L3)>PI THEN 750 ELSE 790
750 IF L1<0 THEN L1=PI+L1 ELSE L1=L1-PI
760 IF L2<0 THEN L2=PI+L2 ELSE L2=L2-PI
770 IF L3<0 THEN L3=PI+L3 ELSE L3=L3-PI
780 FLAG=1
790 IF F1=F2 THEN F1=F1+0.00001
800 IF F2=F3 THEN F2=F2-0.00001
810 IF F1=F3 THEN F1=F1+0.00001
820 REM KONSTANTER
830 A=COS(F1)/(SIN(F1)-SIN(F2))
840 B=COS(F1)/(SIN(F1)-SIN(F3))
850 C=COS(F2)/(SIN(F2)-SIN(F1))
860 D=COS(F3)/(SIN(F1)-SIN(F3))
870 SP=PI/36 :: LG1=L1-PI/2 :: LG2=L3+PI/2 ! SLINGA S
TARTVÄRDE
880 DISPLAY AT(20,14):SP
890 REM BERÄKNINGS-SLINGA
900 FOR L=LG1 TO LG2 STEP SP
910 Z=Z+1
920 X=(A+B)*COS(L-L1)+C*COS(L-L2)+D*COS(L-L3)
930 DISPLAY AT(16,14):L
940 DISPLAY AT(18,14):X
950 DISPLAY AT(12,22):Z
960 N=N+1
970 Y(N)=X
980 IF N<2 THEN 1030
990 K=Y(2)-Y(1)
1000 IF ABS(X)<ABS(K)*5 THEN 1270
1010 IF ABS(X)<1E-7 THEN 1050
1020 IF N>2 THEN N=0
1030 NEXT L
1040 REM TRANSFORM LONG
1050 CALL SCREEN(7):: CALL HCHAR(13,1,32,320)
1060 LG=L*(180/PI)
1070 IF FLAG THEN 1080 ELSE 1090
1080 IF LG<0 THEN LG=LG+180 ELSE LG=LG-180
1090 IF LG>0 THEN L$="O" ELSE L$="V" :: LG=ABS(LG)
1100 LM=(LG-INT(LG))*60
1110 LG=INT(LG)

```

Nya program

av Anders Persson

```
1120 REM LATITUD
1130 X1=(COS(F2)*COS(L-L2)-COS(F1)*COS(L-L1))/(SIN(F1)
-SIN(F2))
1140 X2=(COS(F2)*COS(L-L2)-COS(F3)*COS(L-L3))/(SIN(F3)
-SIN(F2))
1150 FF1=ATN(X1)
1160 FF2=ATN(X2)
1170 F=(FF1+FF2)/2
1180 FG=F*(180/PI)
1190 IF FG<0 THEN F$="S" ELSE F$="N"
1200 FG=ABS(FG)
1210 FM=(FG-INT(FG))*60
1220 FG=INT(FG)
1230 DISPLAY AT(10,2):"T"
1240 DISPLAY AT(10,6):F$;FG :: DISPLAY AT(10,12):USING
"$$. $$":FM !SKRIVER LATITUDEN
1250 DISPLAY AT(10,18):L$;LG :: DISPLAY AT(10,24):USIN
G "$$. $$":LM !SKRIVER LONGITUDEN GOTO 1300
1270 LG=L :: SP=SP/2 :: DISPLAY AT(20,12):SP :: N=0 !
GER SLINGAN NYA VÄRDEN
1280 Y(2)=0 :: Y(1)=0 :: GOTO 900
1290 REM VÄRDEN FÖR DISTANSER
1300 DISPLAY AT(14,7):"DISTANSMIN KM"
1310 DISPLAY AT(16,1):"A-T="
1320 FUT=F1 :: LUT=L1 :: FIN=F :: LIN=L :: GOSUB 1530
1330 DISPLAY AT(17,1):"B-T="
1340 FUT=F2 :: LUT=L2 :: GOSUB 1530
1350 DISPLAY AT(18,1):"C-T="
1360 FUT=F3 :: LUT=L3 :: GOSUB 1530
1370 DISPLAY AT(20,1):"A-B="
1380 FUT=F1 :: LUT=L1 :: FIN=F2 :: LIN=L2 :: GOSUB 153
0
1390 DISPLAY AT(21,1):"A-C="
1400 FUT=F1 :: LUT=L1 :: FIN=F3 :: LIN=L3 :: GOSUB 153
0
1410 DISPLAY AT(22,1):"R-C="
1420 FUT=F2 :: LUT=L2 :: FIN=F3 :: LIN=L3 :: GOSUB 153
0
1430 GOSUB 1630
1440 GOSUB 1700
1450 DISPLAY AT(24,1):"TRYCK: F=FLERA BER. S=SLUT"
1460 CALL KEY(0,KK,SS):: SS=0
1470 IF KK=ASC("F") THEN 330
1480 IF KK=ASC("S") THEN 1490 ELSE 1460
1490 CALL CLEAR :: CALL SCREEN(11):: DISPLAY AT(10,12)
:"S L U T" :: DISPLAY AT(14,12):"19_85"
1500 FOR I=1 TO 800 :: NEXT I
1510 END
1520 REM DISTANSFORMELN
1530 P=P+1 :: Codi=SIN(FUT)*SIN(FIN)+COS(FUT)*COS(FIN)
*COS(LUT-LIN)
1540 DI=ARCCOS(Codi)
1550 DI=DI*(180/PI)
1560 DIMIN=DI*60
1570 KM=DIMIN*1.852
1580 IF KM-INT(KM)>0.5 THEN KM=KM+1
1590 KM=INT(KM)
1600 DIS(P)=DIMIN :: KI(P)=KM
1610 RETURN
1620 REM PRÖVA LONGITUD
1630 IF DIS(1)-INT(DIS(1))>0.5 THEN DIS(1)=DIS(1)+1 ::
FL1=1
1640 IF DIS(2)-INT(DIS(2))>0.5 THEN DIS(2)=DIS(2)+1 ::
FL2=1
1650 IF DIS(3)-INT(DIS(3))>0.5 THEN DIS(3)=DIS(3)+1 ::
FL3=1
1660 IF INT(DIS(1))=INT(DIS(2)) AND INT(DIS(2))=INT(DIS
(3)) AND DIS(1)>5400 THEN 1670 ELSE 1665
1665 IF FL1 THEN DIS(1)=DIS(1)-1 :: FL1=0
1666 IF FL2 THEN DIS(2)=DIS(2)-1 :: FL2=0
1667 IF FL3 THEN DIS(3)=DIS(3)-1 :: FL3=0
1668 RETURN
1670 L=L+PI
1680 IF L>2*PI THEN L=L-2*PI
1690 FLAG=0 :: P=0 :: GOTO 1060
1700 R=16 :: FOR P=1 TO 6 ! UTSKRIVNING DISTANSER
1710 IF R=19 THEN R=R+1
1720 DISPLAY AT(R,6):DIS(P)
1730 DISPLAY AT(R,22):KI(P) :: R=R+1 :: NEXT P :: P=0 :
RETURN
```

KÖPES

32 k Fristående minnesutbyggnad

Björn Landsberg
Fredrikssonsvägen 10
126 54 Hägersten

Sedan i somras har vi jagat efter en del program som vi visste var under distribution från TI:s gömmor och som det ryktades att de skulle ställa till användarföreningarnas förfogande. Den som lyckades blev Anders Persson i Lund. Han öppnade en egen kanal till vännen Maurice T Swinnen, som raskt sände över grejerna.

Tack Anders!

Beställ som vanligt hos föreningen med betalning i förskott. Priset blir 60:- kr per diskett för MULTIPLAN/TI WRITER (en diskett) och DEBUGGER (en diskett) och 120:- kr för FORTH-koden (tre disketter). Vi räknar också med att tillhandahålla FORTH-koden tryckt på papper. Tv säljer vi FORTH-paketet bara till dem som tidigare köpt FORTH från föreningen.
/ Lennart Lindberg

Genom kontakter med Maurice T Swinnen har jag lyckats få tag i några nya program och nya versioner av gamla program. Samtliga program får spridas fritt - de är, som det heter, "in the public domain" - och tillhandahålls av föreningen till ungefär självkostnad i form av program på disketter.

Det handlar om följande:

TI-WRITER

Om någon fortfarande har en version utan riktiga små bokstäver är det här räddningen. Jag har dessutom själv ändrat på teckenuppsättningen efter det att jag fick tag på den här nya versionen. Nu har TI-Writer inte bara små bokstäver utan också riktiga underslängar. Detta gäller dock enbart den svenska versionen.

All ledtext och alla kommando i editorn är på svenska. Utskriftsdelen har också förbättrats en del. Man får filnamnet för skrivaren automatiskt, utan att behöva mata in det varje gång. Dessutom fungerar "Mailing List"-funktionen, nästan... Det går inte som det ska om man försöker mata in variabelistorna från tangentbordet, men om man har den i en fil fungerar det utmärkt.

Lyckligtvis är det sistnämnda den variant som man har absolut störst nytta av. Den andra metoden kan man möjligen använda sig av om det gäller något enstaka brev. Den fungerar just för ett brev i taget. Utskriftsdelen var enbart på engelska från början, men den har jag härjat i rejält. Nu är den enbart på svenska.

Detta gäller både ledtexter och de kommandon som den förstår.

TI-Writer modulen är nödvändig för att kunna använda programmet.

MULTIPLAN

TI/Microsoft Multiplan finns också i en ny version. Den påstås vara avsevärt snabbare, men det vill jag inte hålla med om. Den är snabbare i vissa avseenden, det är riktigt. Dessutom har den fått "auto-repeat", vilket är speciellt trevligt när man ska flytta omkring markören på skärmen.

Även här har jag ändrat på tecknens utseende för att få med underslängar. Har man en dator med definierbar teckenuppsättning kan man ju utnyttja det!

Multiplan modulen samt originaldisketten är nödvändig.

och programversioner

SUPER DEBUGGER

Det här är en debugger från Navarone. Den försöker i mesta möjliga mån likna Texas version som medföljer Editor/Assembler systemet. Skillnaden är att ett antal kommandon har försvunnit och några nya kommit till. Man kan använda båda två tillsammans om man vill, och på så sätt få det bästa av två världar. Det som har tillkommit är möjligheten att disassemblera kod i minnet (men det gör den disassembler som funnits i tidningen bättre), listning av minnesinnehåll på skrivare och flera varianter av enkelstegning.

Dels kan man köra steg för steg på vanligt sätt, dvs man trycker på en tangent mellan varje instruktion. Efter varje instruktion skrivs innehållet i WP, PC och ST ut.

Dessutom kan man köra assemblerprogram interpretativt. Då håller maskinen på tills en viss angiven minnescell når ett specificerat värde. På det sättet får man ett slags "trace" på assemblerprogram. Inte så dumt om inget annat hjälper.

Debuggern kan användas med E/A modulen eller med Extended BASIC. Nyttan med att använda den tillsammans med X-BASIC är dock begränsad, eftersom debuggern ensam fyller nästan hela 8K delen av minnesexpansionen.

KÄLLKODEN TILL TI-FORTH.

Det här är källkoden till hela Forthsystemet, inklusive drivrutiner för I/O etc. Alla ord som TI-Forth ken i grundutförande finns med, definierade i ett assemblerprogram. Dessutom kan man se hur TI utnyttjar de dolda subprogram som finns i styrprogrammet till diskkontrollkortet. Mycket intressant om man vill veta inte bara att datorn fungerar utan även hur det går till. Dessutom är källkoden naturligtvis en guldgruva för alla som är riktigt intresserade av hur Forth fungerar. Källkoden är skriven för Editor/Assemblermodulen.

Annon

SÄLJES

1 st TEXAS TI 99/4A +

Peripheral Expansion System med
Diskdrive + Diskdrive Controller Card,
Memory Expansion card 32 kb RAM,
RS 232 Interface,
Flex Cable Interface.

Moduler:

Disk Manager 2, till diskdrive enheten
TI Extended Basic, TI's Demonstrationsmodul,
Personal Record Keeping,

Alpiner, Indoor Soccer, Multiplication 1,
Number Magic, Parsec, TI Invaders,
Video Chess,

Kassetter:

Teach Yourself Extended Basic,
Electrical Engineering Library,
Graphics Package,

Joystics

Pris 11 000:-

Erling Mattsson Tel: 0660 - 169 26
Ångermanlandsgatan 16 A
891 00 ÖRN SKÖLD SVIK

BESKRIVNING FÖR HANDHAVANDE

av Anders Persson

MULTIPLAN

Den här nya versionen av Multiplan är snabbare i vissa avseenden. Dessutom har den "auto-repeat".

Instruktioner: Byt ut originalfilerna MPBASE, MPCHAR, MPDATA, MPINTR och OVERLAY mot de nya på disketten.

TI-WRITER

Den har nu tecken med riktiga underslängar (OBS: Endast i svensk version, §6). Dessutom har den inledande "form-feed" som utskriftsdelen tidigare gav ifrån sig försvunnit. Förslag till skrivarnamn ges för att underlätta. TI har fortfarande inte fått rätt på Mailing-list funktionen. Det fungerar inte som det ska från tangentbordet, men om man har variabellistan i en fil går det bra.

Instruktioner: Flytta de nya filerna CHARA1 - CHARG1, EDITA1, EDITA2, FORMA1, FORMA2 och TXTEA1 - TXTEG1 till din TI-Writer-diskett.

AVANCERAD DEBUGGER FÖR ASSEMBLERPROGRAM

Den här avlusaren har en del nya funktioner som inte finns i den DEBUG som följer med Editor/Assembler modulen. Filen HELP på skivan förklarar hur programmet fungerar.

KÄLLKODEN TILL TI-FORTH

Källkoden är delad på två disketter.

Del 1: ASMSRC är Forth's dictionary med de ord som finns definierade när TI-Forth startas. Delad i tre filer.

Del 2: DRIVER är koden för I/O systemet, avbrottshantering, SYSTEM-rutiner etc. Dessutom finns i UTIL* delar av de rutiner som laddas av E/A modulen. Dessa finns nämligen på annan plats i minnet när Forth är igång.

Del 2 innehåller även BOOT. När Forth laddats med ASMSRC och DRIVER, kan Forth (när fil-orden laddats) spara en kopia av delar av sig självt på disk. BOOT läser denna kopia och rekonstruerar systemet som filen anger.

När systemet laddas på vanligt sätt, innehåller filen FORTH objekt-koden av BOOT, medan filen FORTHSAVE är kopian av systemet. Om storleken på systemet ändras måste vissa adresser i BOOT modifieras.

Jag har själv modifierat teckendefinitionerna i Multiplan och TI-Writer (svenska versionen) efter det att programmet kom hit.

SÄLJES

Expansionsbox

Roland Pettersson Tel 031-822910
Dr Westrings gata 21 V
413 24 Göteborg

KÖPES

Extended Basic - modul med dokumentation.
Expansionsbox och Disc Control Card.

Ring 0278/40 727. Säkra vardagar efter kl 16.
Fråga efter Stefan.

Stefan Onkenhout, Bollnäs.

LT-WRITER

av Lennart Thelander

Jag har skrivit en liten ordbehandlare, som jag utformat i två versioner.

Till den ena versionen har jag haft stor hjälp av Anders Persson med att skriva assemblerprogram som gör en hel del nyttiga saker. Den första versionen, den enklare, som jag kallar LT-Miniwriter, är avsedd för Extended BASIC och kassettspelare. Den andra, som jag kallar LT-Writer, kräver dessutom minnesexpansionen. Den andra versionen innehåller den första plus en hel del andra nyttigheter.

Båda programmen har en radorienterad editor där texten lagras i en vektor av strängar. Man kan skjuta in rader var som helst (Insert), ta bort rader (Delete), ändra i rader (Change), leta efter en viss text (Locate) och lagra allt på kassett (Save och Read). Det finns en pekare (I) som pekar på den rad som kallas Actline. Det är den pekaren som används till bland annat Insert. Det finns också en variabel (MA) som håller reda på hur många strängar som f.n. är använda. Det andra programmet innehåller en rutin som formatterar om texten till en viss längd och därefter gör högerjustering.

MINIVERSIONEN: KOMMANDON

I(nsert) gör att du kan mata in rad för rad i din text. Om pekaren I pekar någonstans mitt i din text kommer dina nya rader att skjutas in mitt i din text. Här kan du använda Texas' vanliga tangenter (Del, Ins, Erase, pilar) och raden kan bli upp till 255 tecken lång vid inmatningen. För att avsluta skriver du ett CTRL/C först på raden. Då kommer du tillbaka till kommandoläge.

P(rint) utan att radnummer anges skriver alla dina rader på skärmen med radnumren först. När du trycker ned en tangent kommer listan att stanna tills du trycker igen. Om du skriver ett eller två radnummer kommer den eller dessa rader att skrivas utan radnummer på skärmen.

W(rite) skriver alla rader på skrivare utan radnummer.

S(ave) sparar alla rader på kassett. OBSERVERA ATT RADERNA SPARAS TVÅ OCH TVÅ PÅ KASSETTEN OCH ATT DERAS LÄNGD INTE FÅR ÖVERSKRIDA 95 TECKEN. I ANNAT FALL KAPAS DE.

R(ead) läser alla rader från kassett.

M(ove) flyttar pekaren I till det nummer man skriver efter M:et och skriver ut denna rad.

N(ext) flyttar pekaren I till nästa rad och skriver ut denna rad.

X (eXchange) byter skiljetecknet för Change och Locate till det tecken man skriver omedelbart efter X. Detta tecken är från början snedstreck /, men om man vill ändra eller söka efter en text med snedstreck måste man ändra skiljetecknet.

L(ocate) letar upp den text som man skriver efter L mellan två skiljetecken (/) och flyttar pekaren I dit samt skriver ut raden. Se även X. Sökningen börjar på raden efter Actline och går till slutet, därefter från rad 1 till och med Actline.

C(hange) ändrar en text till en annan på den rad man skriver eller på alla rader (då frågar programmet om man vill ändra). Se även X och den hjälptext programmet ger. Om ingen rad anges avses Actline.

D(elete) tar bort en eller flera rader. Programmet listar alla raderna och frågar om det skall ta bort alla rader man begärt. Om ingen rad anges avses Actline.

H(elp) skriver ut den hjälpsida programmet börjar med.

E(xit) avslutar efter en fråga programkörningen.

G(et) tar reda på hur mycket minne som finns kvar: Mindre än 255 bytes eller mindre än 510 bytes eller mer än 510 bytes kvar.

Därmed har Du fått en fullständig förteckning över alla de kommandon mini-versionen innehåller. Normal-versionen kan mera.

MINIVERSIONEN: PROGRAMBESKRIVNING

100-110 Prescan.
120-130 Svenska tecken. CTRL/C
140-150 Initiering: ON ERROR för att programmet inte skall krascha.
160-180 Hör till insert.
190-360 Inläsning och avkodning av kommando.
370-520 Ändra i texten (Change).
530-590 Print med radnummer.
600-620 Insert. Gör plats för en rad i taget.
630-680 Locate. Söker text.
690 Next. Nästa rad.
700 Move. Flytta radpekaren.
710-750 Save. Spara på kassett.
760-810 Read. Läs från kassett.
820 Avluta programmet.
830-850 Write. Skriv på printer. OBS! Kolla att filnamnet i OPEN passar för din printer.
860-930 Help. Skriver hjälptext.
940-980 Print. Skriv alla rader på skärmen.
990 Xchange. Ändrar skiljetecken.
1010 ON ERROR rutin.
1020-1130 Delete. Tar bort rader.
1140-1160 Get. Kollar hur mycket minne som är kvar.
1170 Används av Read och Save.

MINIVERSIONEN: KOMMENTARER

Det här programmet kan inte så mycket, huvudsakligen beroende på att det tar plats i minnet att lära programmet mer. Det går emellertid att utöka programmet ganska enkelt genom att ta valda delar ur nästa program och ersätta de nuvarande raderna. Strunta i de ensamma CALL LOAD som finns där; de finns där för användarvänlighetens skull men är inte nödvändiga. Om någon rutin innehåller ett CALL LINK kan den inte användas i det mindre programmet, emellertid.

NORMALVERSIONEN: UPPSTART

När det större programmet startar ska det läsa data från kassetten. Dessa data innehåller teckendefinitioner och även en del assemblerprogram som är till stor nytta. Teckendefinitionerna är ASCII-tecken enligt svensk standard och innehåller riktiga små bokstäver inklusive underslängar. Denna datafil skapas av det program som kommer härnäst på kassetten. Det programmet läser in en fil med teckendefinitioner som finns efter det programmet, ändrar en del tecken och skapar den fil som LTWRITER läser in. Om det är några tecken som behöver ändras kan det göras innan det sista CALL LINK i det programmet. Kom bara ihåg att alla tecken är uppflyttade en pixel för att göra plats för underslängar. Den här versionen av programmet fungerar i stort som den första versionen, men med en del tillägg. Här kommer en liten 'bruksanvisning'.

NORMALVERSIONEN: KOMMANDON

Alla gemensamma kommandon fungerar nästan lika i båda versionerna. Det är bara Read och Save som skiljer.

ORDbehandling

I(nsert) se ovan och Format.

C(hange) se ovan.

P(rint) se ovan.

L(ocate) se ovan.

N(ext) se ovan.

M(ove) se ovan.

S(ave) fungerar helt annorlunda. Här finns ingen begränsning på stränglängd utan alla strängar sätts ihop till ett (två) block som skrivs ut på kassetten i ett (två) svep. Det går MYCKET fortare än i den första versionen.

R(ead) se Save. Om du redan har text i datorn kan du få de inlästa strängarna att skriva över den gamla texten eller fylla på efteråt. Jag har inte testat vad som händer med programmet om minnet blir fullt i detta läge. Om man vill kan man sedan flytta strängarna med Buffer och Output. Därefter kan man ta bort dem från den plats där de hamnade från början med Delete.

E(xit) se ovan.

W(rite) se ovan.

D(elete) se ovan.

H(elp) se ovan.

X(change) se ovan.

G(et) se ovan.

B(uffer) kopierar en eller flera strängar till en buffert. OBS! Radnummer måste anges. Om detta göres flera gånger kommer de nya strängarna att läggas efter de gamla. OBS! Formattering kommer att ändra på innehåll i bufferten. Bufferten har plats för 300 strängar (lika många som den vanliga texten).

O(utputs buffer) Raderna i bufferten kommer att skjutas in före Actline. Bufferten tömmas efteråt.

Z(eroes buffer) Tömmer bufferten (Innehållet går förlorat).

F(ormat) Detta programs största fördel. Med detta kommando får man texten högerjusterad till den radbredd man vill ha (28 till 132 tecken; se rad 1550 i listningen). Formatteringen göres i två steg. Det första innebär att texten formas om så att längden på strängarna blir mindre än eller lika med maxbredden. Om en sträng är för lång kommer alltså en del av orden att hamna i nästa sträng och om en sträng är för kort flyttas nästa sträng upp. Programmet fyller raderna tills nästa ord inte får plats. Om ett ord är längre än maxbredden kommer ordet att avstavas (utan hänsyn till några grammatiska regler). Det andra steget innebär att ett antal blanktecken skjuts in jämnt mellan orden så att rätt radlängd erhålls. Det sistnämnda kommer inte att ske vid slutet av ett stycke eller vid slut på formatteringen. Nytt stycke markeras med en helt tom rad. Nästa rad börjar då indragen två tecken och raden ovanför kommer inte att fyllas med blanka till maxbredden.

Om man av en eller annan anledning vill att en del av texten inte skall formatteras (tabeller eller annat), kan det också göras. Man skriver då ett kontrolltecken i texten (CTRL/E). Då kommer det tecknet först på en rad i texten och efterföljande text kommer att vara likadan som när man skrev in den. Om man vill att formatteringen skall börja på nytt, skriver man in ett CTRL/F i texten. Det kontrolltecknet kommer då sist på den sista rad som inte formatterades. Därifrån börjar formatteringen som vanligt igen. OBSERVERA att alla kontrolltecken man matar in i texten kommer att finnas kvar (även CTRL/E och CTRL/F). Kontrollera att er printer ignorerar dessa tecken (ASCII 133 och 134). Anledningen till att kontrolltecknen inte borttages är att man kanske vill formattera igen efter en ändring. Av denna anledning kommer alla överflödiga blanka att tas bort i fas ett så att man kan justera texten riktigt i fas två.

Det var hela bruksanvisningen. Jag hoppas att jag inte har glömt något. Alla synpunkter är välkomna till mig.

NORMALVERSIONEN: PROGRAMBESKRIVNING

100-110 Prescan.
120-130 Initiering.
140 Stänger av QUIT-tangenten.
150-350 Assemblerprogram för inläsning av fil från kassett.
360 Anrop av nämnda assemblerprogram.
370 Mer initiering.
380-390 Initiera Save och Read, Actline=1, hoppa till Hjälpssidan.
400-420 Hör till insert.
430-640 Inläsning av kommando, därefter avkodning och hopp till rätt programdel.
650-800 Change.
810-870 Print utan radnummer på skärmen.
880-900 Insert. Hör samman med 400-420.
910-960 Locate.
970 Next.
980 Move.
990-1050 Save. Se nedan.
1060-1150 Read. Se nedan.
1160 Exit.
1170-1190 Write. Kolla att rad 1170 passar till din printer. Om skrivaren klarar mer än 80 tecken per rad, måste även maximal radlängd anges. Exempel: OPEN \$1: "Filnamn", VARIABLE 136 Skrivare som tar liggande A4 kan kanske skriva ännu längre rader.
1200-1270 Help.
1280-1320 Print. Skriver all text på skärmen med radnummer.
1330 eXchange.
1340 Används av Move och Next bl.a.
1350 Felåterhämtning
1360-1470 Delete.
1480-1500 Buffer.
1510-1530 Outputs buffer.
1540 Zeroes buffer.
1550-1730 Format; Formatterar utan högerjustering. Flyttar CTRL/E och CTRL/F till riktiga positioner på raden.
1740-1760 G Kollar hur mycket minne som finns kvar.
1770-2010 Format fortsättning. Fyller i blanka där det behövs.

SÄRSKILDA FÖRKLARINGAR

HÄR skall jag försöka beskriva alla CALL LOAD och CALL LINK som utföres när programmet exekveras. Jag börjar med LOAD-satserna. På rad 430 och rad 890 finns 'CALL LOAD(-31884, 5)'. Denna rad sätter tangentbordet i läge 5, dvs stora och små bokstäver samt kontrolltecken. På rad 440 laddas i stället en trea. Det betyder att man inte längre kan mata in små bokstäver, utan de blir stora. På rad 1160 finns en LOAD som hjälper till att stanna programmet så att inget händer inför nästa körning (Assemblerprogrammen är känsliga för omstarter).

Lista över LINK-anrop:

"FLIP" startar en interruptdriven cursorflip.

"NOFLIP" stannar densamma (Flippen tar ca 5% av kapaciteten).

"STINIT" Initierar lagring av strängar i minnesexpansionens 8 kilobytes del. Detta används för att spara strängarna snabbare på kassett. Som parameter används en variabel som håller reda på adressen där nästa sträng kan lagras. I detta program heter variabeln 'START'.

"STFREE" ger antalet bytes lediga för stränglagring. Har två parametrar: 'START' och 'BYTES'. Den senare innehåller efter anropet antalet lediga bytes.

"STCLR" rensar minnet från gamla strängar. 'START' användes som parameter.

"STSTO" lagrar en sträng i 8K delen. Parametrar:
'START' och strängen som skall lagras.

"STRCL" hämtar en sträng från 8K delen. Parametrar:
'START' och en strängvariabel där strängen lagras.

"STSAVE" lagrar de strängar som finns i 8K delen på den
fil vars namn finns som parameter (t.ex "CS1" eller
möjligen "DSK1.STRINGS").

"STLOAD" hämtar strängar från fil till 8K delen.
Filnamn som parameter. Observera att 8K delen först
måste tömmas med STCLR. Annars kommer de nya
strängarna att lagras efter de gamla i 8K delen. Om
minnet då blir fullt blir det fel (I/O ERROR med en
felkod som är felaktig). Om minnet räcker kan man på
detta sätt sätta ihop texter med hjälp av flera filer
(görs på annat sätt i LTWRITER).

Man måste veta hur många strängar man har innan man gör
alla STRCL för att lagra strängarna på vanligt sätt.
Assemblerprogrammet läser först en byte som talar om
hur lång strängen är och tar sedan så många bytes och
lagrar som tecken i strängar. Om man gör STRCL för
många gånger kan man få de mest konstiga strängar. I
LTWRITER har jag löst det på följande sätt: Först
lagrar jag som sträng det antal strängar som jag sedan
skriver. Därefter lagrar jag så många strängar som
rymms i 8K delen (om det finns så många). När minnet är
fullt lagrar jag som sista sträng en bestående av tre
kontrolltecken, vilka inte kan matas in från
tangenterbordet. Detta block spelas sedan in på kassetten.
Om det finns flera strängar kvar lagras dessa i 8K
delen i nästa omgång, för vidare befodran till
kassetten. I 8K delen kan lagras c:a 7000 bytes med
strängar. Två block om 7000 bytes är 14000, dvs mer än
de maximalt 13928 bytes minne där strängar lagras (VDP
RAM).

Sedan när jag skall läsa in strängarna läser jag ett
block från kassetten. Där står först hur många strängar
som finns. Därefter flyttar jag strängarna så länge det
finns fler eller tills jag hittar strängen med
kontrolltecken. Om den strängen fanns, läser jag in
ett block till och flyttar därefter över strängarna
till VDP RAM som vanligt.

Jag vill passa på att tacka Anders Persson för hans
eminenta assemblerprogram. Jag skulle inte kunna
skriva dem själv ens om jag hade någon assembler.

Eftersom det är väldigt jobbigt att själv knappa in det
här programmet med den fil som programmet läser, hade
jag tänkt göra så här: Den som vill ha programmet kan
skicka en tom kassetten, ett frakterat och adresserat
kuvert till mig tillsammans med en tia för besväret,
varvid jag spelar in alla nödvändiga filer på
kassetten. Den som inte har råd med tio kronor får ta
bort en sats i programmet som skapar den fil som
LTWRITER läser in. Den sats som skall tas bort heter
'CALL LINK("LOADCH", "CS1")'. Sedan är det bara att
göra så många CALL CHAR man önskar för att få riktiga
små bokstäver. Kom ihåg att alla tecken är flyttade
uppåt en pixel för att få rum med underslängar. Den
som har disksystem kan ändra filnamnet i den sista
satsen i nämnda program till t.ex DSK1.LTSUPPORT. Då
måste man också ändra sats 360 i LTWRITER till 'CALL
LINK("SUPPORT", "DSK1.LTSUPPORT")'. Den som har
disksystem ändrar lämpligen också på filnamnen i Save
och Read till lämpliga namn. Det allra bästa vore att
läsa in filnamnet till F\$ och skriva CALL
LINK("STSAVE", F\$) och motsvarande för STLOAD.

Den som vill ha en kassetten hittar mig på denna adress:

Lennart Thelander
Ultunagatan 46
252 62 Helsingborg

LT-MINIWRITER/THELANDER

```
100 GOTO 120 :: DIM A$(300):: CALL CLEAR :: CALL KEY
:: CALL SCREEN :: CALL CHAR :: K,S,MA,I,F,W,E,R=0
:: S$,D$,F$,Q$,C$
110 CALL SOUND :: Q,T,Y,U,MARGIN :: !'P-
120 CALL CHAR(91,"00280038447C44440028007C4444447C001
00038447C44444"):: CALL CHAR(131,"00304030060806")
130 CALL CHAR(123,"0000280038447C44000028007C44447C00
00100038447C44")
140 CALL CLEAR :: CALL SCREEN(4):: C$="/" :: ON ERROR
1010
150 I=1 :: GOTO 860
160 MA=MAX(MA,I):: Q$=A$(I):: PRINT STR$(I)&">";: AC
CEPT BEEP:A$(I)
170 IF LEN(A$(I))=0 THEN I=I+1 :: GOTO 610 ELSE IF AS
C(A$(I))<>131 THEN I=I+1 :: GOTO 610
180 A$(I)=Q$ :: FOR F=I TO MA :: A$(F)=A$(F+1):: NEXT
F :: MA=MA-1
190 CALL KEY(5,K,S):: LINPUT "CMD>":Q$ :: IF Q$>"Z" T
HEN Q$=CHR$(ASC(Q$)-32)&SEG$(Q$,2,255)
200 CALL KEY(3,K,S)
210 IF SEG$(Q$,1,1)="C" THEN 370
220 IF SEG$(Q$,1,1)="I" THEN 600
230 IF Q$="P" THEN 940
240 IF SEG$(Q$,1,1)="P" THEN 530
250 IF SEG$(Q$,1,1)="L" THEN 630
260 IF Q$="N" THEN 690
270 IF SEG$(Q$,1,1)="M" THEN 700
280 IF Q$="S" THEN 710
290 IF Q$="R" THEN 760
300 IF Q$="E" THEN 820
310 IF Q$="G" THEN 1140
320 IF Q$="W" THEN 830
330 IF SEG$(Q$,1,1)="D" THEN 1020
340 IF Q$="H" THEN 860
350 IF SEG$(Q$,1,1)="X" THEN 990
360 PRINT "ILLEGAL COMMAND" :: GOTO 190
370 W=POS(Q$,C$,1)
380 E=POS(Q$,C$,W+1)
390 R=POS(Q$,C$,E+1)
400 IF W=0 OR E=0 OR R=0 THEN PRINT "WRONG FORMAT" ::
GOTO 190
410 S$=SEG$(Q$,W+1,E-W-1)
420 D$=SEG$(Q$,E+1,R-E-1)
430 F$=SEG$(Q$,R+1,LEN(Q$)-R)
440 IF F$="" THEN F=I :: GOTO 510 ELSE IF F$<>"ALL" A
ND F$<>"all" THEN F=VAL(F$):: GOTO 510
450 FOR F=1 TO MA
460 W=POS(A$(F),S$,1):: IF W=0 THEN 490
470 PRINT STR$(F)&">";A$(F):: INPUT "CHANGE THIS Y/N?
":Q$ :: IF Q$<>"Y" THEN 490
480 A$(F)=SEG$(A$(F),1,W-1)&D$&SEG$(A$(F),W+LEN(S$),L
EN(A$(F))-W+LEN(S$)):: PRINT STR$(F)&">";A$(F)::
GOTO 460
490 NEXT F
500 GOTO 190
510 I=F :: W=POS(A$(I),S$,1):: IF W=0 THEN PRINT "STR
ING "&S$;" NOT FOUND" :: GOTO 190
520 A$(I)=SEG$(A$(I),1,W-1)&D$&SEG$(A$(I),W+LEN(S$),L
EN(A$(I))-W+LEN(S$)):: PRINT A$(I):: GOTO 190
530 W=POS(Q$,",",1)
540 IF W=0 THEN I=VAL(SEG$(Q$,2,LEN(Q$)-1)):: PRINT A
$(I):: GOTO 190
550 E=VAL(SEG$(Q$,2,W-2)):: R=VAL(SEG$(Q$,W+1,LEN(Q$)
-W+1))
560 FOR F=E TO R :: PRINT A$(F)
570 CALL KEY(0,K,S):: IF S=0 THEN 590
580 CALL KEY(0,K,S):: IF S>>1 THEN 580
590 NEXT F :: GOTO 190
600 PRINT "TYPE CTRL/C IN POS 1 TO EXITTYPE CTRL/E TO
END FORMAT TYPE CTRL/F TO START FORMAT"
610 CALL KEY(5,K,S):: IF I>MA THEN 160
620 FOR F=MA TO I STEP -1 :: A$(F+1)=A$(F):: NEXT F ::
A$(I)="" :: MA=MA+1 :: GOTO 160
630 S$=SEG$(Q$,3,LEN(Q$)-3)
640 FOR F=I+1 TO MA :: GOSUB 670 :: NEXT F
650 FOR F=1 TO I :: GOSUB 670 :: NEXT F
660 PRINT "NO LOCATION" :: GOTO 190
670 W=POS(A$(F),S$,1):: IF W=0 THEN RETURN
680 I=F :: GOTO 1000
690 I=I+1 :: I=MIN(I,300):: MA=MAX(MA,I):: GOTO 1000
700 I=VAL(SEG$(Q$,2,LEN(Q$)-1)):: I=MIN(I,300):: MA=M
AX(MA,I):: GOTO 1000
```

ORDbehandling

```

710 PRINT "SAVE": "CHANGE CASSETTE TAPE"
720 OPEN $1:"CS1",OUTPUT, FIXED 192, INTERNAL
730 PRINT $1:MA :: GOSUB 1170
740 FOR F=1 TO MA STEP 2 :: PRINT $1:SEG$(A$(F),1,95)
,SEG$(A$(F+1),1,95):: NEXT F
750 CLOSE $1 :: GOTO 190
760 PRINT "READ" :: PRINT "CHANGE CASSETTE TAPE"
770 OPEN $1:"CS1",INPUT, FIXED 192, INTERNAL
780 E=MA :: INPUT $1:MA :: GOSUB 1170
790 FOR F=1 TO MA STEP 2 :: INPUT $1:A$(F),A$(F+1)::
NEXT F
800 FOR F=MA+1 TO E :: A$(F)=" " :: NEXT F
810 CLOSE $1 :: GOTO 190
820 INPUT "DO YOU WANT TO EXIT Y/N? ":Q$ :: IF Q$<>"Y
" THEN 190 ELSE STOP
830 OPEN $1:"RS232.BA=9600.DA=8" :: INPUT "LEFT MARGI
N>":MARGIN
840 FOR F=1 TO MA :: PRINT $1:TAB(MARGIN);A$(F):: NEX
T F
850 CLOSE $1 :: GOTO 190
860 CALL CLEAR
870 PRINT "P print all lines": "G gets memory space"
880 PRINT "Px<,y> print line x<thru y> L/text/ locate
s text C/text1/text2/(x?ALL) change tex
t1 to text2 on line x or on all lines"
890 PRINT "S saves all lines on CS1 R reads all li
nes from CS1 Mx moves to line x N next
line"
900 PRINT "I inserts before actline E exits progra
m W writes on printer DX<,y>
del's line x<thru y>"
910 PRINT "H help"
920 PRINT "X_ changes delimiter char to any cha
r( )"
930 GOTO 190
940 FOR F=1 TO MA :: PRINT STR$(F); ">"; A$(F)
950 CALL KEY(C,K,S):: IF S=0 THEN 970
960 CALL KEY(C,K,S):: IF S>1 THEN 960
970 NEXT F
980 GOTO 190
990 C$=SEG$(Q$,2,1):: GOTO 190
1000 PRINT STR$(I)&">"&A$(I):: GOTO 190
1010 ON ERROR 1010 :: PRINT "BAD COMMAND" :: RETURN 19
0
1020 W=POS(Q$,"",1):: IF W>0 THEN 1080
1030 IF Q$<>"D" THEN I=VAL(SEG$(Q$,2,LEN(Q$)-1))
1040 PRINT STR$(I); ">"; A$(I): "DO YOU WANT TO DELETE TH
IS <Y/N>";:: INPUT Q$ :: IF Q$<>"Y" THEN 190
1050 FOR F=I+1 TO MA :: A$(F-1)=A$(F):: NEXT F
1060 IF MA=I THEN A$(MA)=" " :: MA=MA-1 :: MA=MAX(MA,1
):: I=MIN(MA,I)
1070 GOTO 190
1080 E=VAL(SEG$(Q$,2,W-2):: R=VAL(SEG$(Q$,W+1,LEN(Q$)
-W+1):: W=R-E :: IF E>MA OR R>MA OR W<=0 THEN 36
0
1090 FOR F=E TO R :: PRINT STR$(F); ">"; A$(F):: NEXT F
1100 PRINT "DO YOU WANT TO DELETE THIS <Y/N>";:: INPU
T Q$ :: IF Q$<>"Y" THEN 190
1110 FOR F=R+1 TO MA :: A$(F-W-1)=A$(F):: NEXT F
1120 FOR F=0 TO W :: A$(MA-F)=" " :: NEXT F
1130 MA=MA-W-1 :: MA=MAX(MA,1):: I=MIN(I,MA):: GOTO 19
0
1140 ON ERROR 1150 :: Q$=RPT$( "F",255):: ON ERROR 1160
:: S$=Q$ :: Q$,S$=" " :: PRINT "MORE THAN 510 BYT
ES LEFT" :: ON ERROR 1010 :: GOTO 190
1150 Q$=" " :: ON ERROR 1010 :: PRINT "LESS THAN 255 BY
TES LEFT" :: GOTO 190
1160 Q$,S$=" " :: ON ERROR 1010 :: PRINT "LESS THAN 510
BYTES LEFT" :: GOTO 190
1170 W=MA/60*4 :: PRINT "PLEASE WAIT AT LEAST":INT(W)
;"MIN"S";(W-INT(W))*60;"SEC"S" :: RETURN

```

Fortsätter i PB 85-2 med

LT-SUPPORT

PROGRAMBITEN 85-1

COMPUTERBLADET

"Ti'eren"



TRÆF 1985

Ti'eren No. 2 April/Maj 1985.

Lørdag den 11. og Søndag den 12. maj bliver det første "TI 99 COMPUTERTRÆF" afholdt i Danmark. Vi har tidligere nævnt træffet i Lund i Sverige, men vi håber at dette træf vil blive af en helt anden dimension. Det vil blive kørt over to dage, hvor der begge dage vil blive mange ting at se og høre fra flere grene af programmeringsteknikken. Der vil blive flere komplette anlæg, der vil komme til at køre forskellige programmer og sprog, f.eks. TI-BASIC -- EKT-BASIC -- MINI MEN og ASSEMBLER -- TERMINAL EMULATOR med et MODEM dels med et modul og dels med et diskette program foruden TALE med SYNTHESIZER og WYCOVE FORTH v 2.1 for ikke at glemme EKT-BASIC II der er en spændende og ny måde at lave grafik og mange andre ting.... Træffet bliver holdt på kursus-centret "HØJGÅRD" Bagsværd Hovedgade 274 . 2880 Bagsværd. Det er det lokale Højvang Radio, Bladsaxe, der er vært og arrangerer for dette træf, senere er det meningen at andre lokale klubber skal være værter så vi kommer ud til dem der måtte ønske det. Der går busser og tog næsten lige til døren, sådant: Tog til Skovbrynet station ca 5 min. gang fra os Bus 63 fra Rådhuspladsen til Værebros parken Bus 165 fra Lyngby til Bagsværd hovedgade. Bus 43 fra Centrum City til Værebros Parken. Det er også nemt at finde i bil Centret ligger nær hvor Hillerød motorvejen krydser Ring IV Sø for Hareskov. For dem der kommer langvejs fra og ønsker at overnatte, er det muligt at disponere over 20 sengepl. på Centret, til en pris af Kr.185.- incl. Morgenmad Det vil blive muligt at købe mad og drikke på stedet. Da det af hensyn til mad og overnatning er nødvendigt med et forvarsel bedes i sende et af de reserveringskort der er hæftet i bladet, man kan tilmelde sig klubvis eller som enkeltperson efter eget ønske. Samtidigt skal det oplyses at det desværre ikke er muligt at lave et sådan træf uden at det koster penge og der er derfor en entre på Kr.20.- pr. person, børn u. 14 år halv pris. Yderligere oplysninger kan indhentes hos enten: Peter Colmorten . På tlf. (01) 69 73 48 eller host Chr. Sørensen . På tlf. (03) 75 65 53

K-TRYCK AB
HASSELIQUIST. 39 . 121 46 JOHANNESHÖV

KONSTIGA TECKEN

av Lennart Lindberg

Jag vill återkomma till temat om teckenrepresentation på olika ASCII-koder. Det var uppe i nr 84-4, sid 16. Den grund-teckenstandard som t ex olika programmeringsspråk använder är förstås den amerikanska. A-et i ASCII betyder just "American". Den varierar sedan i olika länder världen över. Framförallt är det 12 tecken som varierar. Jag listar här bredvid ett antal sådana varianter, sådana de är utformade av den stora skrivare-tillverkaren EPSON. EPSON har genom sin dominans på marknaden åstadkommit vad som brukar kallas en "de facto"-standard. Det innebär att EPSONS lösningar betraktas som standard utan att det är fastställt formellt. Varianterna är den amerikanska (kallad USA) den svenska (kallad SWEDEN) den ena danska (kallad DENMARK1)

ASCII-KODER OCH MOTSVARANDE TECKEN: EPSON RX
MATRISKRIVARE, TECKEN-'SET' USA, 12 TKN/TUM
35 36 64 91 92 93 94 96 123 124 125 126
\$ % & ' () * + , - . / : ;

ASCII-KODER OCH MOTSVARANDE TECKEN: EPSON RX
MATRISKRIVARE, TECKEN-'SET' SWEDEN, 12 TKN/TUM
35 36 64 91 92 93 94 96 123 124 125 126
\$ % & ' () * + , - . / : ;

ASCII-KODER OCH MOTSVARANDE TECKEN: EPSON RX
MATRISKRIVARE, TECKEN-'SET' DENMARK1, 12 TKN/TUM
35 36 64 91 92 93 94 96 123 124 125 126
\$ % & ' () * + , - . / : ;

Den danska är med därför att den representerar en bra kompromiss mellan skandinaviska tecken och de tecken man behöver för programmering. De är alla utskrivna med en matris skrivare, där det är lätt att ändra teckenutseende genom en programmerad instruktion från ditt program till den lilla dator som sitter i skrivaren. Problemet med matris skrivare är att de inte ger särskilt klar och tydlig skrift - särskilt inte vid tryckning av t ex en tidning. Vi vill hellre använda sk skönskrivare, som numera arbetar mest med typhjul. Vi har i det här numret använt framförallt en skönskrivare av märket Diablo med ett typhjul med typsnittet ELITE - en välkänd gammal skrivmaskinstil för 12 tecken per tum. På det typhjul vi har ser de 12 tecknen ut som framgår av listningen här

ASCII-KODER OCH MOTSVARANDE TECKEN: DIABLO 1355WP
SKÖNSKRIVARE MED TYPHJUL ELITE, 12 TKN/TUM
35 36 64 91 92 93 94 96 123 124 125 126
\$ % & ' () * + , - . / : ;

Vi har också använt en skönskrivare av märket SENSE med ett typhjul som kan skriva såväl typsnittet PICA med 10 tecken per tum som ELITE. Den skrivaren är använd till en del FORTH-skärmar. Dess 12 tecken finns också listade här

ASCII-KODER OCH MOTSVARANDE TECKEN: SENSE P01
SKÖNSKRIVARE MED TYPHJUL ELITE, 12 TKN/TUM
35 36 64 91 92 93 94 96 123 124 125 126
\$ % & ' () * + , - . / : ;

Varför nu denna utläggning? Därför att varje skrivare blir en kompromiss mellan olika intressen - snabbhet i utskrift, teckenuppsättning, läsbarhet, snygghet osv. Det innebär att ett visst missnöje alltid kan uppstå med valet. Jag vill bara ha litet förståelse för det, gärna också synpunkter. Sedan tror jag att det här kan vara intressant och lärorikt också. Jag har i varje fall tyckt det när jag alltmer börjat begripa hur det hänger ihop.

Ett par detaljer till:

- Diablo-hjulet vi använt saknar tecknet alfa-slang, som är viktigt i Forth. För att kunna skriva FORTH-spalten med Diablon har jag valt att sätta in ett annat tecken i stället, nämligen %. Det går ju så lätt att göra det med en ordbehandlare. Den "blip" som

sitter där alfa-slang skulle ha suttit på hjulet är för otydlig, tror jag. Den uppmärksamme ser nu också snabbt att alfa-slang saknas även på SENSE-hjulet. Där det skulle ha suttit finns i stället ett paragraf-tecken, vilket alltså skall läsas som alfa-slang (fetch) på FORTH-skärmarna som skrivits med SENSE. På Diablo-hjulet finns också ett paragraf-tecken, men det sitter på den plats där "number sign" (bräddgård) finns i amerikansk standard. Det innebär att listningar av Basic och X-Basic innehåller paragraf-tecken i t ex IMAGE och i referenser till filer och sprites. - i TI-Writer används tecknet på ASCII-koden 94 (shift 6, kallad circumflex eller tak) som en styrsignal till formatteringsprogrammet. Jag har använt TI-Writer till någon artikel av Jan Alexandersson, där ett Basic-kommando för "upphöjt till" förekommer. Det är just det tecknet. I listningen blev det blankt, vilket alltså inte berodde på skrivaren - det syns ju här, där jag skriver tecknet mellan parenteserna () - utan på ordbehandlaren. Tecknet är ditritat för hand nu i Jans artikel.

Nästa utvecklingssteg på den typografiska fronten är att skapa ett program som gör det möjligt att utnyttja Diablon förmåga till proportionell skrift. Vi kan då få fram en tidning med vackrare och jämnare stil och rak högermarginal samtidigt. Vi får se hur det går med det! Kanske vi kan byta till ett hjul med en fiffigare teckenuppsättning på också.

Upprop PASCAL

Ett upprop av Anders Persson

Du som har TI:s Pascal - pcode-kort och hela rasket. Jag vill ha kontakt med Dig.

Jag gör numera helst mina program i Pascal när jag inte jobbar i assembler. Tyvärr har jag inte hittat så många att prata erfarenheter med - Pascal verkar inte vara så spritt bland 99-folket. Jag har hittat Lars Skyttner i Jämtland och Bengt Fahlgren i Stockholm. Jag vet att det finns någon mera. Kanske det också finns för mig helt okända Pascal-intresserade?

Du som är Pascalintresserad - tag kontakt med mig! Jag vill ha flera samtalspartners. Vi kan säkert få till något roligt och bra. Jag har sträckt ut tentaklarna mot England t ex. Kanske en internationell PIG: Pascal Interest Group for TI 99.

Jag finns som vanligt i Lund, på Kämnärsvägen 4:1078, 222 45 Lund. Telefonnummer är 046-11 69 19.

Pascal-vänner! Föreningen Er!

PROGRAMBANKEN

91 ** SPEL
02911002E --X Prickskytte med kanon mot en kyckling !
02911003E --- Startrek, textadventure
02911006E --- Othello.
02911007E --- Robotjakt.
02911008E --- 15-spel.
02911009E --- Tärningsspel.
02911011E --- Keno
03911013E --+D Starguard
03911014E --- Miner
03911015E --C Yahtzee
03911016E --- Backgammon
03911017E --X 3D Tic-tac-toe
03911018E --X Not one
03911019E --X Datorpoker
03911024E --- Fyra i rad (luffarschack)
04911025E --- Startrek 2
04911026E --X Hjälp kycklingen över vägen
04911027E --- Katapult
01911028H --- Car driver
01911029H --- Animals
06911032E --- Vem skjuter den sista roboten ?
06911034H --X L-game
07911035H --X Wari
07911036H --- Ta dig fram i en osynlig labyrint.
07911037 --X Eliza

07911039H --X Teckenjakt
07911040H --X Black jack
07911041 --X Space battle
07911043 --- Enarmad bandit
08911039E --X Sprite-jakt
08911046E --X Damspel.
08911049E --X Swords & sorcery, textadventure
08911051E --X Ägg-fångst
09911052E --- Deep space
09911054H --- Planetary lander.
09911056H --X Lunar lander.
09911057F --- Towers of Hanoi.
09911058E --X Breakout, bollspel squashtyp.
09911059H --J Ritprogram i olika färger.
09911060H -JX Treasure hunt, labyrintspel.
10911061H --X Kermit
10911062H -JX Find the gun
10911063E --- Matematikspel.
10911065E --- Camel, äventureritt i öknen
10911008E --- Colorfraktions, matematikspel
06912004E --- Slalom
06912005E --- Killer
09912008F --X Asteriod (tal).
09912009 --- Battlestar
11911055S --X Hängning
11917001S -JX Inkräktarna
11917002S -JX Pärön

FÖRENINGENS

TILLBEHÖRS- FÖRSÄLJNING

Följande finns att köpa för medlemmar genom att mot-
svarande belopp sätts in på postgiro 19 83 00 - 6.

Användartips med Mini Memory	60:-
FORTH Olika versioner,	
Föreningens Forth	Kassett eller diskett 250:-
II-Forth	Diskett 60:-
II-Forth manual	190:-
Nittinian I-tröja	40:-
99'er Magazine nr 12/82, nr 1-5,7-9/83 (per styck)	20:-
Nittinian, årgång 1983	80:-
Programbiten - nittinian årgång 1984	100:-
Programbiten, årgång 1983	Kalkylatorer 60:-
Programbiten, årgång 1982	60:-
Programbiten, årgång 1981	60:-
Programbiten, årgång 1980	60:-
Programbiten, årgång 1978/79	60:-
Programbiten, fem årgångar 1978-1983	200:-
Programbiten, sex årgångar 1978-1984	280:-
Katalog med belgiska och engelska program för räknare TI-57, TI-58, TI-59	20:-
Föreningens programmeringsblanketter (TI-59), olika typer, block om 50 blanketter (se PB 83-1 sidan 30) per block	11:-
Patenthandlingar TI-59	25:-
40 st tomma magnetkort med plånbok	150:-
Tom magnetkortsplånbok	10:-

DATAVISION MODUL

Modulen gör att du kan kommunicera med DATAVISION- eller liknande baser.

Utskrift för printer finns tillgänglig, sparning av skärmar på kassett eller diskett.

Nödvändig utrustning:

RS232 interface

Televerkets modem eller liknade

Modulen kommer med instruktioner om hur du kopplar upp dig med Televerkets modem.

Pris: 565:-

16 KRAM CMOS MODUL

utan hölje. Modulen kan simulera ett ROM, inladdning sker från I.ex FORTH vilken medför att minimum utrustning är X-basic, MINIMENORY, eller ED/ASM och 32 K minnesutökning. Ram'et är batteribackupat så programmet ligger kvar minst 2 veckor. Denna modul används i princip för att testa ut hur egna program i modulform fungerar innan man bränner prommar vilket blir klart jobbigare.

Pris: 875:-

