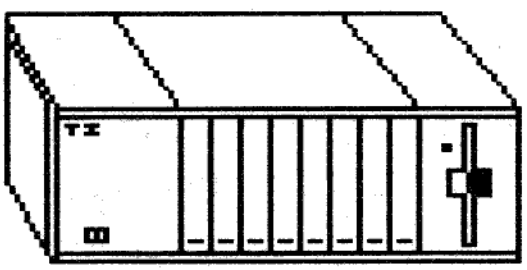
PROGRAM nittinian BITEN 85-3

FORTH spalten

REDAKTÖR: Lars-Erik Svahn



DU SKA FÖRSÖKA PLACERA UT SVENSKA STÄDER PÅ KARTAN HÄRINTILL

MED STYRSPAKEN STYR DU RINGEN TILL STADENS LÄGE PÅ KARTAN

HÅLL DEN RÖDA KNAPPEN INTRYCKT SÅ GÅR DET SNABBARE

TRYCK ENTER

Innehåll

ISSN 0281-1146

Redaktören...

Det här är det första numret av Programbiten som jag ansvarar för. Som vanligt har det tagit en viss tid att sätta sig in i rutinerna. Tidningen har ju (åtminstone i mitt tycke) genomgått en klar förbättring de senaste åren. Att nå upp till samma nivå som mina företrädare blir inte lätt.

Det här numret ser lite annorlunda ut än de tidigare. Merparten av artiklarna ha passerat en fotosättmaskin innan de gått till tryck. Gången har varit att artiklarna kommer in till redaktionen på olika media och skapade med olika program. Disketter som går att läsa med TI-Writer stavningskollas med ett program som Anders Persson beskriver i detta nummer. När all text finns på diskett förs den över till en speciell diskettläsare på ett sätteri och därifrån kablas den över till sätteridator. Eftersom detta råkar vara mitt dagliga värv, belastar inte sättningen föreningens ekonomi. En förhoppning från min sida är att merparten av de problem med teckenuppsättningar på olika skrivare som förekommit, nu snart ska vara ett minne blott. Synpunkter på tidningens läslighet mottages tacksamt.

Någon brist på material har det inte varit till detta nummer. Det har tex kommit in ovanligt mycket recensioner, både på program och tillbehör. Antagligen är detta ett resultat av allt det som rör sig runt 99-an ute i världen. Men det finns alltid plats för mer artiklar och framförallt program. Som vanligt gäller att ju bättre ett program är kommenterat från författaren, ju lättare blir arbetet för redaktören.

Det kommer en hel del hårdvarutips till redaktionen, det mesta från olika utländska kontakter. Det finns ett behov av någon som kan sovra i materialet och anpassa det till komponenter som finns tillgängliga i Sverige osv. Kort sagt, jag tror vi behöver en "hårdvaruredaktör", precis som vi har det på andra områden. Arbetet är förstås ideellt som allt annat arbete i föreningen, men säkert stimulerande för dem som kan något om elektronik. Hör av dig på tidningens adress om du känner dig manad!

Vi efterlyser också fler artiklar om hur just *du* använder din 99:a. Vi vet att det finns massor av medlemmar som har stor nytta av datorn i sitt arbete eller hobby. Skriv några rader och berätta! Har du något foto som illustrerar vad du gör, desto bättre.

Den tävling om nyttoprogram som utlysts tidigare i Programbiten har tyvärr inte hittills gett många svar. För att ge fler medlemmar chansen kommer tiden att förlängas till den 15 september 1986. Du har alltså sommaren på dig att göra klart programmet du börjat på.

Peter Odelryd

Möte i Stockholm

Boka in lördagen den 13 september för en medlemsträff i Stockholm. Lokalen blir den sedvanliga, aulan i Militärhögskolan, Valhallavägen 117. T-bana till Stadion. Tid: kl 14.00–18.00. Temat för mötet blir de nya program och tillbehör som kommit den senaste tiden till 99-an. Vi kommer bla att visa *Maximem* och en del freewareprogram som presenteras i detta nummer, tex c99 och screendumpprogram. Det kommer att finnas möjlighet att köpa freewareprogrammen på diskett.

Välkommen till en som vi hoppas givande eftermiddag!

Ett ytterligare möte är inbokat till lördag den 29 november, samma tid och plats. Till detta återkommer vi senare.

I redaktionen:

Redaktör	Peter Odelryd
Utmaningsredaktör	Anders Persson
Forth-redaktör	Lars-Erik Svahn
Programförmedlare	Börje Häll
Allt-i-allo	Claes Schibler

Föreningens och redaktionens adress:
Föreningen Programbiten
c/o Schibler
Wahlbergsgatan 6, 1 tr ned
12146 Johanneshov

Datainspektionens licensnummer: 82100488

Postgiro: 198300-6
Medlemsavgiften för 1986 är 120:–

Föreningens tillbehörsförsäljning

Följande finns att köpa för medlemmar genom att motsvarande belopp insätts på postgiro 198300-6.

För TI-99/4A:	
Användartips med Mini Memory	60:–
PB-Forth kassett (föreningens Forth)	250:–
PB-Forth diskett (föreningens Forth)	250:–
PB- och TI-Forth+TI-Forth manual vid *samtidig* beställning (600 för icke medlem)!!	400:–
TI-Forth (diskett)	60:–
TI-Forth (manual)	190:–
PB-Forth källkod (3 disketter) [1]	120:–
TI-Forth källkod+demo (2 disketter) [2]	120:–
Game-Forth diskett	60:–
Forth&99 (3 disketter) [1]	120:–
Disassembler (diskett)	60:–
Multiplan/TI-Writer uppgraderingar diskett	60:–
Super Debugger diskett	60:–
Datavision modul	565:–
16 K RAM CMOS modul	875:–
Nittinian T-tröja	40:–
99'er magazine nr 12/82, 1–5, 7–9/83 (per styck)	20:–
Programbiten/Nittinian 1985 (TI 59+99)	100:–
Programbiten/Nittinian 1984 (TI 59+99)	80:–
Nittinian, årgång 1983 (TI-99/4A)	60:–
För TI-59 mm	
Programbiten, årgång 1983 (kalkylatorer)	40:–
Programbiten, årgång 1982 (kalkylatorer)	40:–
Programbiten, årgång 1981 (kalkylatorer)	40:–
Programbiten, årgång 1980 (kalkylatorer)	40:–
Programbiten, årgång 1978/79 (kalkylatorer)	40:–
Programbiten, fem årgångar 1978–83	160:–
Programbiten, sex årgångar 1978–84	220:–
Programbiten, sju årgångar 1978–85	300:–
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 sid 30) per block	11:–
Patenthandlingar TI-59	25:–
40 st tomma magnetkort med plånbok	SLUT
Tom magnetkortsplånbok	10:–

[1] Säljes endast till den som tidigare köpt PB-Forth
[2] Säljes endast till den som tidigare köpt TI-Forth

Föreningen Programbiten Årsmötesprotokoll

Stämman 1986-02-08

Mötet ägde rum i Militärhögskolans samlingssal i Stockholm den 8 februari 1986 med början kl 14.00.

1. Mötet öppnades av föreningens ordförande Lennart Lindberg. De närvarande kontrollerades mot medlemsregistret av Hans Wickström.

2. Till ordförande för mötet valdes Lennart Lindberg, till sekreterare Hans Wickström och till justeringsmän Gunnar Lenke och Bo Nordlin.

3. Mötet förklarades vara behörigen utlyst.

4. Dagordningen godkändes sådan den presenterats i kallelsen.

5. Styrelsens årsredovisning 1985.
– Verksamhetsberättelsen presenterades
– Balans- och Resultaträkningen presenterades
Mötet beslöt att godkänna årsredovisningen.

6. Bo Carleö föredrog revisorernas berättelse. Revisorerna föreslog att styrelsen för 1985 skulle beviljas ansvarsfrihet.

7. Fråga om ansvarsfrihet för styrelsen.
Styrelsen beviljades ansvarsfrihet för sin förvaltning sådan den framgått av lämnad redovisning av mötet.

8. Stadgar.
De stadgeändringar som togs upp på förslag under årsmötet 1985 fastställdes. D v s de stadgar som finns att läsa i programbiten 1/85 gäller härmed.

9. Verksamheten 1986.
Verksamheten 1986 diskuterades. De bör enligt mötets mening innehålla följande:

Tidning, från 4 nr på 28 sidor till 5 nr på 24 sidor men dock under 1986 6 nummer. Redaktionsdeadline 31/3, 31/5, 30/9, 15/11.

Träffarna föreslogs att även i fortsättningen vara på MHS. Lennart Lindberg ställde sig ej ovillig till förslaget.

Vårmöte, kallelse i samband med tidning.

Teman; Dataprogram och nya produkter.

Databas, prel databas?

Distribution ur programbanken.

Mikrovisionen: Distribution av program via Datavision.

Mikrodatorns anslagstavla

Internationella kontakter.

Spridning i landet.

Kontakt med 99-ägare på Volvo.

10. Val av styrelse för 1986.
Valberedningen föredrog sitt förslag. Därefter valdes;
– till ordförande Jan Alexandersson
– till sekreterare Hans Wickström
– till kassör Mikael Nordlin
– till redaktör Peter Odelryd
– till övriga ledamöter:
Börje Häll, programförmedlare
Tony Rogvall
Claes Schibler
Anders Törnqvist
Michael Öhman

11. Val av revisorer.
Till revisorer valdes Gunnar Lenke och Barbro Nygren.
Till revisorssuppleant valdes Bo Carleö.

12. Val av valnämnd.
Till valberedning valdes Göran Nygren och Lennart Lindberg.

13. Årsavgiften fastställdes till 120 kronor. Medlemsantalet under 1984 var ca 500 och under 1985 ca 360.

14. Allmänt medlemsmöte.
Börje Häll presenterade resultatet av enkäten, 115 st medlemmar hade svarat och resultat kommer att presenteras i 3/86.

Peter Odelryd pratade om tidskriften Micropendium.

SDF (Sveriges Data Förbund) diskuterades

Programbanken. Föreningen har från USA fått tillgång till nya s k Freeware:
– Diskmanager
– Screen Dump
– Neat List
genom Peter Odelryds försorg.

15. Inga beslut fattades under denna punkt.

16. Inga beslut fattades under denna punkt.

17. Mötet avslutades.

Vid protokollet

Hans Wickström

Hans Wickström

Justeras

Jan Alexandersson

Jan Alexandersson

Gunnar Lenke

Bo Nordlin

Bo Nordlin

Ordföranden ...

Det är nu dags för den nya styrelsen att ta över arbetet. Först ett stort tack till Lennart Lindberg, Gösta Blume och Göran Nygren som lämnar styrelsen.

Lennart Lindberg har som ordförande gjort en bra insats när det gäller att inspirera andra till insatser och även fixat en mycket bra möteslokal. Lennart har även bidragit med ett stort antal artiklar till tidningen, i första hand om FORTH men även intressanta saker om nyheter och 128k RAM mm.

Gösta Blume har gjort ett gediget arbete som kassör.

Göran Nygren har varit redaktör under 1984. Det var då som tidningen blev mer innehållsrik. Under senaste året har Göran varit programförmedlare.

Det pågår ganska stor aktivitet ute i världen när det gäller att ta fram nya program och tillbehör till TI-99/4A främst i USA och Västtyskland men även Kanada bör nämnas.

I USA har *Millers Graphics* kommit med *GRAM-kracker, Advanced Diagnostics* och *Explorer*. Alla dessa kan användas för att gräva sig in i alla detaljer i maskin och operativsystem. Han kommer även med ett kort för att kunna ansluta IBM PC-kompatibelt tangentbord till TI-99/4A.

Myarc har kommit med ett 128K och 512K RAM-kort samt en ny snabbare Ext Basic som använder sig av detta 128k-kort. Deras helt nya dator tycks dock dröja.

Texcomp säljer ut TI originalprogram till mycket låga priser som *LOGO II* för USD 19,95, *TI-Writer* för USD 38,95, *Multiplan* för USD 38,95, *Editor-Assembler* för USD 19,95 och *Mini Memory* för USD 38,95.

I Västtyskland har det kommit ett tyskt expansionssystem, *GRAM-karte* som har vissa likheter med GRAM-kracker men placeras i expansionsboxen, *Mega-RAM* en RAM-disk på 1 Mbyte, 80-kolumners kort med den nya videoprocessorn 9938, Assemblator för XB+EM med CS eller DSK, GPL-Assembler.

Från Kanada kommer *Maximem, Companion, Disk Manager 1000* och *Small-C*. De två senare är freeware och kommer att distribueras till medlemmarna.

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

För att underlätta för den hårt prövade redaktionen, eller kanske snarare redaktören, kommer jag i fortsättningen inte att skriva om en massa ovidkommande saker i Utmaningen. Blir det inga problem – eller lösningar – blir det ingen Utmaning heller. Punkt.

Automatisk avlusning

Det här låter väl som en dröm för många som har lagt pannan i djupa veck över tredskande program. Tyvärr är rubriken en sanning med modifikation. Det är nämligen inte de program du själv skriver det gäller. Men jag vet att några haft problem med program som funnits i tidningen. Dessa program, som – i regel – bevisligen fungerar, har ihärdigt vägrat att utföra sin uppgift. Detta trots att du noga kontrollerat att de är korrekt inmatade. Lösningen brukar vara att du trots allt inte har hittat de fel du gjort vid inknappningen.

Inom redaktionen och styrelsen, och även på andra håll, har det förts en diskussion om hur program ska tryckas i tidningen för att de ska vara så lätta som möjligt att läsa. Programmet BUGOUT (finns i anslutning till Utmaningen) tar ett annorlunda grepp på det här problemet. För varje rad i programmet räknar BUGOUT fram en kontrollbokstav. Tanken är att programmen i tidningen ska tryckas tillsammans med dessa bokstäver. När du skrivit av programmet kan du utnyttja BUGOUT till att se om du får samma bokstäver. Får du det är det antagligen rätt. I annat fall är det garanterat fel. Lämpligen börjar du med att kontrollera BUGOUT självt . . .

Ett problem med programmet är att det inte fungerar med mindre än Extended BASIC och skivminne. BUGOUT är skrivet i X-BASIC, och programmet som ska kontrolleras måste först listas på diskett. Jämför med det listprogram som Bo Nordlin skrev (se nummer 84/1).

Med anledning av detta får ni följande utmaning: Skriv ett program som gör samma sak, men som klarar sig utan skivminne.

Såvitt jag kan se ger detta bara ett realistiskt alternativ beträffande val av språk – assembler. Minimal utrustning blir då antingen X-BASIC och minnesexpansion eller TI-BASIC och Mini Memory. Programmen blir naturligtvis olika i de båda fallen.

Tanken är alltså att assemblerprogrammet ska finnas i minnet samtidigt som BASIC-programmet. Kontrollbokstäverna ska genereras direkt, utan mellanlagringar av programmet. Det här kräver naturligtvis kännedom om hur BASIC lagrar program i minnet. Det kan du skaffa dig genom att läsa artiklarna som handlar om Merge med kassettbandspelare i tidigare Utmaningar.

Det BUGOUT som finns publicerat här skapar kodbokstaven med ledning av de tecken som ingår i listan. Dessa finns inte i minnet, eftersom PRINT, IF, CALL osv representeras med olika koder. Såvitt jag kan se finns det inget som hindrar att beräkningen sker direkt med dessa koder. Resultatet blir naturligtvis inte likadant, men det har ju ingen betydelse om de publicerade programmen kodas på samma, avvikande sätt.

Efter det här går vi nu över till lösningarna. Det handlar den här gången om kommersiella lösningar på problem som vi tidigare diskuterat inom Utmaningen.

Korrekturläsning

Någon gång (strax efter Hedenhös eller så) skrev jag att det vore roligt med en automatisk korrekturläsning till TI-Writer. Noga räknat var detta i nummer 84-1.

Jag har nu fått tag på en sådan, om än inte från någon i föreningen Programbiten. Programmet heter Auto Spell-Check och kommer från Dragonslayer American Software Co., Omaha, USA.

Hur fungerar det då? Jo tack, ganska bra faktiskt. Programmet, som alltså körs från alternativ tre i TI-Writer, läser den text som ska kontrolleras. Alla ord i texten jämförs med en ordlista. Om det finns något ord i texten som saknas i listan tillfrågas användaren, som förhoppningsvis vet besked, hur ordet ska stavas.

Lite mer i detalj går det till så här. Först anger man vad filen man vill kontrollera heter. Efter inläsning ber programmet om disketten med ordlistan. När den lästs igenom håller programmet reda på alla ord i texten som inte finns med i ordlistan. Om man så vill kan man nu låta programmet läsa extra listor med specialuttryck. Lagrade ord som finns i dessa extra listor stryks ur minnet.

När alla extra listor lästs, visas de resterande ord som inte kunde identifieras. Här ges flera möjligheter. För det första kan man få se vilken mening ordet står i, om man inte blir klok på vad det är för ett ord. När man vet detta kan man korrigera stavningen om den är felaktig eller godkänna ordet om det är korrekt. Riktiga ord kan antingen lagras i ordlistan eller bara godkännas tillfälligt.

Typiska kandidater till den tillfälliga kategorin är speciella namn och liknande som knappast återkommer. Ord som är någorlunda vanliga lagrar man för framtida bruk. På så sätt lär sig programmet mer och mer efterhand, och blir med tiden en expert på en viss persons ordförråd.

I USA levereras programmet, bland annat av Tex Comp (USD 34,95), med en färdig ordlista innehållande de vanligaste orden, ungefär 20000 ord. Den här listan är naturligtvis oanvändbar för svenska artiklar. Jag har inte tillgång till den överhuvudtaget. Emellertid har jag, genom att utnyttja programmets förmåga att lära sig nya ord, byggt upp en svensk ordlista. Jag lät helt enkelt kontrollera ett antal artiklar som jag skrivit tidigare (Utmaningar bland annat). I början blev det naturligtvis en massa frågor om okända ord, men nu fungerar det ganska bra. Jag har också valt att lägga specialuttryck med datoranknytning i en separat ordlista. Därigenom undviker jag en del onödiga kontroller i mer normal text.

Ett problem här är att programmet är utvecklat i USA, och följaktligen inte tror att Å, Ä och Ö är bokstäver. Jag har ju inte tillgång till källkoden, varför modifieringar är svåra att göra. Jag har i alla fall lyckats övertala programmet att godkänna våra speciella bokstäver inuti ord. Fortfarande vill det dock inte befatta sig med ord som börjar med någon av dessa bokstäver. Varför begriper jag inte. Än så länge . . .

När jag ändå var igång passade jag på att ordna med inläsning av filen med teckendefinitioner som TI-Writer normalt använder. Annars fick jag den vanliga uppsättningen som TI är så förtjusta i. Det är inte jag.

Ordlistans storlek begränsas i princip av tillgängligt skivutrymme. De ordlistor jag byggt upp innehåller för närvarande ungefär 5000 ord, vilket upptar drygt 40 kilobytes på disketten. Programmet tillåter dock att huvudordlistan delas på två filer, vilka alltså kan finnas på olika disketter. Om man då, som jag, har 360 kilobytes per diskett kan man alltså få en ansenlig lista. Vidare finns det ingen begränsning beträffande hur många egna listor man kan använda. I princip kan man alltså kolla mot hur många ord som helst. Praktiskt sett är det naturligtvis så att ett oändligt antal ord tar en oändlig tid att kontrollera, vilket knappast är tilltalande.

Går det då något snabbare än traditionell korrekturläsning? Tja, det beror naturligtvis på hur fort och exakt man själv läser. Jag tog tiden på en kontroll av en text innehållande drygt 16000 tecken. Det motsvarade två och en halv sida i tidningen när artikeln publicerades. Hela proceduren med kontroll mot den normala ordlistan (då drygt 3000 ord) och mot min speciella dataordlista (ca 500 ord) följd av rättelser av felstavningar och uppdatering av ordlistan tog knappt tolv minuter. Det är ungefär lika lång tid som det tar för mig att läsa artikeln. Jämförelsen haltar dock, eftersom programmet arbetar själv ungefär halva den tiden. Dessutom hittade det några fel som jag inte lade märke till. Vidare rättar datorn automatiskt texten, när den fått reda på hur ordet skulle stavats. Efter en genomgång är texten alltså både kontrollerad och rättad!

Hittills har jag inte råkat ut för artiklar som varit för långa för programmet. De största jag provat med omfattar drygt 21000 tecken, vilket är ganska nära gränsen för vad TI-Writer kan hantera. Om det finns väldigt många nya ord i texten kan man naturligtvis råka ut för att programmet inte klarar av att hålla ordning på allihop. Därför är en innehållsrik ordlista väsentlig för en strömlinjeformad funktion.

Det finns, i vanlig ordning, även en baksida på medaljen. Om man gör ett stavfel så att ordet förändras till ett annat giltigt ord, som finns i listan, kommer det att slinka med obemärkt. Någon semantisk eller grammatisk kontroll sker alltså inte. Dessutom är det ju upp till maskinisten att tala om hur nya ord ska stavas. Speciellt i det här fallet, när jag själv var tvungen att bygga upp ordlistan från början, stavar programmet alltså inte ett dugg bättre än vad jag gör. Men när jag väl angett rätt (förhoppningsvis) stavning fångas alla felslag på tangenterna.

Huvudordlistan blir ganska stor efter ett tag. Programmet klarar inte av att lägga till nya ord till hur stora listor som helst. Normalt ska man ju inte lägga till nya ord till något annat än de egna speciella listor som man skapar. Antalet ord i dessa bör vara högst 2000. Men när det gäller uppbyggnad av en grundordlista får man snart en rejäl lista. För att klara detta skrev jag ett eget program som samsorterar två listor. Uppdatering kan då ske till en liten lista, vilken slås ihop med den större med jämna mellanrum.

På det hela taget är jag ganska förtjust i det här programmet. Hädanefter blir naturligtvis alla artiklar som jag skriver kollade med Spell-Check. Om stavfel förekommer i fortsättningen är det alltså helt och hållet maskinens fel . . .

Trace på assemblerprogram

Att kunna se vad som händer i ett assemblerprogram är ett önskemål lika gammalt som assemblerprogrammeringen. I PB 85-1 önskade sig Folke Andersson en sådan möjlighet av det lite mer avancerade slaget. Ett program som gör ungefär det här finns nu att köpa. Det heter Explorer och kommer från Millers Graphics. OBS! Förväxla nu inte detta med "the Explorer" från Tex-Comp. Det är nämligen något helt annat. En recension finns nog någonstans i tidningen, även om jag inte vet det säkert när jag skriver det här.

Terminalemulator

Sådana har vi också diskuterat tidigare. Jag har nu lyckats lägga vantarna på ett program som heter Fast-Term. Det är ett program som, för en gångs skull, har lyckats kombinera de flesta bra finesserna från andra program. Det handlar här om "free-ware", ett påfund som behandlats tidigare i tidningen.

Diskettrutiner

Program som fungerar bättre än Disk Manager modulerna har också sett dagens ljus på senare tid. Disk Manager 1000 är ett av dessa. Även det här är free-ware. De funktioner som finns är ungefär de samma som CorComps disk manager till deras dubbeldensitets kontrollkort. Ett bra tillägg är möjligheten att läsa textfiler direkt från programmet. Den funktionen är en självklarhet i alla vettiga system, men har först nu börjat förekomma på vår kära apparat. Undantag gäller för Forth, som normalt inte har några filer alls, och för Pascal, som skriver ut vilka filer som helst var som helst.

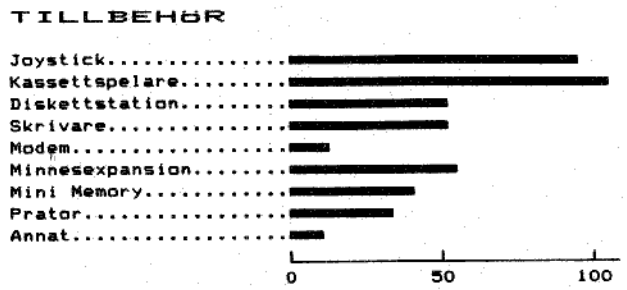
Slut i rutan!

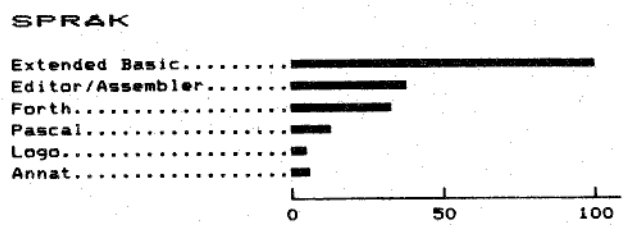
Det här är programmet som genererar kontrollbokstäver. Bokstäverna till vänster är skapade med programmet självt.

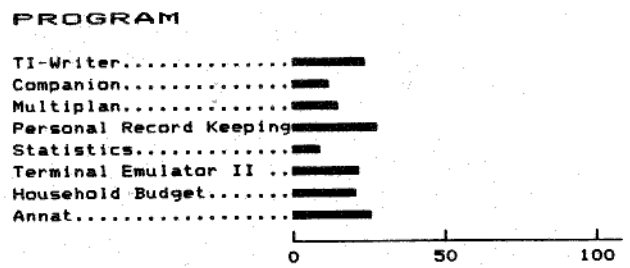
```
P 100 REM ***********
Q 110 REM * BUG-OUT *
P 120 REM ***********
M 130 REM COPYRIGHT 1985
H 140 REM EMERALD VALLEY PUBLISHING CO
A 150 REM BY THE HCM STAFF
R 160 REM HOME COMPUTER MAGAZINE
G 170 REM VERSION 5.5.1
T 180 REM TI EXTENDED BASIC ONLY
D 190 SP$=CHR$(32) :: CR$=CHR$(13)
X 200 DISPLAY AT(2,5)ERASE ALL:"HCM BUG-OUT PROGRAM"
N 210 DISPLAY AT(4,1):"YOU MUST ""LIST"" THE PROGRAM
      TO DISK." :: DISPLAY AT(6,1):"THEN RUN ""BUG-OU
      T"""
P 220 DISPLAY AT(8,1):"FILE NAME: DSK1." :: ACCEPT AT
      (8,12)SIZE(-15):FL$
F 230 ON ERROR 520
R 240 OPEN #2:FL$,INPUT ,DISPLAY ,VARIABLE 80
Q 250 DISPLAY AT(8,1):"SEND OUTPUT TO: 1. SCREEN" :: D
      ISPLAY AT(11,17):"2. PRINTER"
M 260 DISPLAY AT(12,1):"1" :: ACCEPT AT(12,1)VALIDATE
      ("12")SIZE(-1):DV :: DV=DV-1
C 270 IF DV=0 THEN 310
T 280 ON ERROR 530
T 290 DISPLAY AT(13,1):"PRINTER:" :: DISPLAY AT(14,1)
      :"RS232" :: ACCEPT AT(14,1)SIZE(-28):DV$
W 300 OPEN #1:DV$,OUTPUT
N 310 ON ERROR STOP
Z 320 IF EOF(2)THEN 500
A 330 LINPUT #2:C$
E 340 IF C$="" THEN 320
Y 350 N=POS(C$,SP$,1):: OT$=SEG$(C$,1,N):: LNNUM=VAL(
      OT$):: IF LEN(C$)<80 THEN 400
A 360 FLAG=1 :: IF EOF(2)THEN FLAG=0 :: GOTO 400 ELSE
      LINPUT #2:D$
Z 370 M=POS(D$,SP$,1):: IF M>5 THEN 390
Y 380 ON ERROR 540 :: LN=VAL(SEG$(D$,1,M)):: IF LN>LN
      NUM THEN 400
Y 390 C$=C$&D$ :: FLAG=0
M 400 CK$=C$ :: M1=POS(C$," REM ",N):: M2=POS(C$," !
      ",N):: IF M1<1 AND M2<1 THEN 450
A 410 IF M1=N THEN CK$=SEG$(C$,1,N+4):: GOTO 450
A 420 IF M2=N THEN CK$=SEG$(C$,1,N+2):: GOTO 450
H 430 IF M1>N THEN CK$=SEG$(C$,1,POS(C$,":: REM ",1)+
      4 :: GOTO 450
Y 440 IF M2>N THEN CK$=SEG$(C$,1,POS(C$,":: ! ",1)+2)
W 450 CK=0 :: CK1=0 :: FOR I=1 TO LEN(CK$):: CHK=ASC(
      SEG$(CK$,I,1)):: CK1=CK1+CHK :: IF I/2=INT(I/2)
      THEN CK=CK-CHK ELSE CK=CK+CHK
M 460 NEXT I
C 470 CK=ABS(CK)*CK1 :: I=INT(CK/26):: CK=CK-I*26
W 480 OT$=OT$&SP$&CHR$(65+CK)
S 490 PRINT #DV$:OT$ :: IF FLAG=0 THEN C$="" :: GOTO
      320 ELSE C$=D$ :: FLAG=0 :: GOTO 350
A 500 CLOSE #2 :: PRINT "ALL DONE" :: IF DV=1 THEN CL
      OSE #1
H 510 END
H 520 DISPLAY AT(10,1)BEEP:"FILE NOT FOUND" :: GOSUB
      550 :: GOTO 200
P 530 DISPLAY AT(14,1):"PRINTER NOT RECOGNIZED" :: GO
      SUB 550 :: RETURN 280
X 540 CALL ERR(EN,ET,ES,EL):: IF EN=74 THEN RETURN 39
      0 ELSE PRINT "ERROR NUMBER: ";EN:"IN LINE ";EL
      :: STOP
M 550 FOR DE=1 TO 500 :: NEXT DE :: RETURN
```

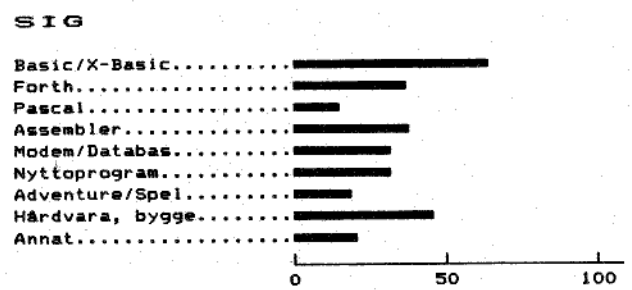
Enkäten

Här kommer en sammanställning av enkäten som medlemmarna fick svara på under 1985. Det har inkommit 115 svar och dessa redovisas nedan. Först visas stapeldiagram för svaren på de fyra första frågorna. Därefter kommer en kortfattad sammanställning av de svar, som inte går att rita staplar på, dvs annat under de fyra första frågorna och frågorna under "vad vill du säga om tidskriften". Sist kommer en del synpunkter från enkäten. Frågor från densamma kommer att besvaras under rubriken Frågor och Svar. Enkäten kommer att vara vägledande för styrelsen i det fortsatta arbetet för föreningen.









Tillbehör
Under annat tillbehör återfinns bla 128 Kb minne, Bar Code Reader och klocka.

Programmeringsspråk
Här finner vi ASPIC och Super Extended Basic.

Nyttoprogram
54 svar har vi under annat nyttoprogram. Här nämner vi bara en del programtyper: Egna program, Ekonomi, Math Routine Lib, Ordbehandling och registerhantering. Under varje programtyp finns fler olika program.

SIG
Under SIG annat kan nämnas följande: Ekonomi, Grafik, Kalkyl, Kommunikation, Naturvetenskapliga beräkningar, Ordbehandling, Registerhantering och Undervisningsprogram.

Jag vill läsa mer om
Presentationer/recensioner, Hårdvara, Frågespalten, Grafik och ljud, Kommunikation, Register, Program/programmering, Tips och trix, Utmaningen, Assembler, Basic, Forth, Pascal.

Jag vill läsa mindre om
Avancerade saker utan enkla förklaringar, Programlistningar, Recensioner, Spel, Assembler, Basic, Forth, Pascal.

Jag vill fortsätta läsa om
Assembler, Forth, Pascal, X-Basic, Hårdvara, Mini Memory, Nyttoprogram, Ordbehandling, Program/programmeringstips, Recensioner, Register, Spel, Tips och trix, Utmaningen.

I övrigt vill jag ha sagt
Beskrivning av pinnarna i PIO, Bättring beträffande programbanken, Dela tidningen i Basic, assembler osv, Forth borde ha mer plats i tidningen, Forth-skola, Fler möten. Bättre fart på programbanken, Fler nr ca 12/år varför ej en prenumerationsavgift för en sådan ökning, Föreningen alldeles för stockholmsfixerad, Hjälp med programlistningar, Information om utrustningsmöjligheter och inköpsställen, Jobba på, Lista innehållet i promarna i expansionsenheten, Många är inte så informerade i dataslangen, Nyckel att få TI-Basic från andra dialekter, Programmeringsskola i X-Basic, Somliga artiklar för avancerade, Styrelsen sköter sig utmärkt, Svag reaktion på insända brev, Svårighetsgraden för hög, Telefonnr bör ingå i matrikeln, Telefonnr till artikelförfattaren, Tidningen mycket bra, Utmaningen mycket bra.

Synpunkter från enkäten

Enkät: Programutbyten bland medlemmar
Angående ovanstående önskemål anser styrelsen att föreningens medverkan i programförmedling ska ske genom programbanken. Däremot står det fritt för medlemmarna att genom insändare, annons etc ta kontakt med varandra för programbyten.

Enkät: Tips hur man programmerar, hur man gör enkla program, råd programmering.

För att föreningen ska kunna komma med tips och råd måste medlemmarna sända in frågor. Ett forum för detta är spalten *Frågor och Svar*. Om svaret blir för långt kan det kanske resultera i en artikel. Om frågeställaren sänder med ett adresserat och frankerat kuvert så kommer föreningen att, snarast möjligt, returnera svaret, dessutom kommer svaret att tas in i tidningen om frågan inte är för speciell.

Enkät: Medlemmarna bor ej i Stockholm. Ordna möten på andra orter så att fler kan närvara.

Som styrelsen ser det är det två saker som förhindrar styrelsen att ordna möten på andra orter.

För det första är det praktiska problem bl a hyra av lokal (styrelsen har inte kännedom om lämplig lokal, kontaktpersoner mm) samt att ordna med utrustning för demonstrationer av olika slag.

För det andra är det kostnaden för resor, uppehälle, eventuell övernattning mm, föreningens ekonomi tillåter inte detta,

Vad styrelsen kan göra är att hjälpa till med vissa saker beträffande arrangemangen såsom utsändning av kallelse, bidraga med program för demonstrationer och annat som i princip inte kostar något samt att eventuellt bidraga till lokalhyra och förtäring typ kaffe med kaka.

Enkät: Knyckt/saxat ur andra tidningar.

Copyright lägger hinder i vägen för att knycka/saxa ur andra tidningar. Som exempel kan nämnas att sekreteraren, Hans Wickström, har varit i kontakt med en tysk tidning för att få publicera en av deras artiklar, som han har översatt. De ville ha 150 Dmark för att föreningen skulle få publicera artikeln i Programbiten. Om föreningen inte har tillstånd att publicera en artikel från en tidning, finns det risk för stämning och skadestånd.

Explorer

av Anders Persson

Hjälpmedel för felsökning i program, så kallade avlusare (eng debuggers), har funnits länge till alla sorters datorer. Till 99:an fanns i början det modifierade TIBUG programmet som medföljer Editor Assemblermodulen. Dessutom finns det en enklare avlusare inbyggd i Mini Memory.

Nu finns mer avancerade hjälpmedel att tillgå. Explorer från Millers Graphics är ett program som har det mesta man kan tänka sig i den här vägen. Med hjälp av detta får man ett slags "fönster" in i datorns allra innersta. Explorer visar vad datorn håller på med i varje ögonblick. Du kan få reda på processorns status, minnesinnehåll i RAM, VDP RAM och GROM – eller GRAM om man har det.

Programkörningen kan enkelt påverkas från Explorer. Det går att köra ett steg i taget, mycket långsamt, lite fortare, ännu fortare och så vidare. Brytpunkter kan sättas var som helst i programmet, även i ROM. Dessutom kan man säga stopp exempelvis när processorn skriver till en viss adress i VDP RAM.

Inte nog med detta. Explorer har också extra funktioner, normalt dolda, som tillåter exempelvis beräkningar med hexadecimala eller binära tal, läsning och skrivning till CRU bitar och laddning av VDP register.

Den stora finessen med programmet är att det är så lätt att använda och utnyttja. Samtidigt visas information om var i minnet processorn för närvarande exekverar, vilken VDP RAM och GROM adress som senast refererats till, vad workspacen innehåller, vad som finns i minnet på något utvalt ställe och vilken instruktion som ska exekveras härnäst. Allt på en skärm alltså, och dessutom ytterligare några andra saker, faktiskt.

Explorer kan naturligtvis också visa hur skärmen skulle ha sett ut om Explorer inte varit insyltad.

Tekniken

Hur är nu detta möjligt? Jo, det fungerar så att Explorer interpreterar den maskinkod som man vill köra. Detta innebär alltså att processorn inte direkt kör programmet, utan läser av opkoden som data och sedan utför motsvarande operationer. Det är samma sak som när du använder BASIC. CPU:n förstår inte BASIC, men ett översättningsprogram (interpretator) läser BASIC-koden och utför motsvarande operationer. Skillnaden är att det här är ett assemblerprogram som tolkar ett assemblerprogram. Finessen blir att det går att helt kontrollera exekveringen av det program som till synes körs. Egentligen är det ju ett annat program, dvs Explorer, som processorn kör. Jämför med BASIC, där du kan trycka på FCTN-4 och stanna programmet. Datorn stannar inte för det, utan fortsätter lika intensivt med den maskinkod som den alltid arbetar med.

Explorer ger alltså (minst) samma möjligheter vid assemblerprogrammering som annars erbjuds vid arbete med BASIC eller liknande. Priset för detta är snabbhet. Bruksanvisningen anger att Explorer normalt kör datorn med en hastighet som är ungefär 1/300 av den normala. I värsta fall kan hastigheten gå ner till 1/1000 av det normala. De som tror att 99:an är långsam i vanliga fall skulle se den då!

Nu är det här inte så farligt som det låter. Vanligen utnyttjas programmet till att inspektera korta, intressanta delar av ett större program. Om du använder Explorer som avlusare är det då på program som är skrivna direkt i assembler, och alltså innehåller förhållandevis få instruktioner för att utföra en viss uppgift.

För vem?

Som jag ser det passar Explorer till tre olika behov, vilka mycket väl kan förenas i en och samma person.

För det första är Explorer utmärkt till att undersöka vad ens egna assemblerprogram egentligen håller på med. Visserligen är Explorer självt ganska stort (18 K), men det hindrar inte att man testar sansat stora egna rutiner. Genom att dela sina program och utnyttja Mini Memory får man ändå 18 K över till det egna programmet. Om du dessutom har Maximem eller liknande står ju ytterligare massor av minne till förfogande.

För det andra kan Explorer med fördel användas av de som är intresserade av hur 99:an fungerar internt. Det hörs ju på namnet också (eng. för upptäckare). Det finns alla möjligheter att studera hur BASIC, Extended BASIC eller andra moduler egentligen fungerar.

För det tredje bör, såvitt jag kan bedöma, Explorer vara idealisk för dig som verkligen vill lära dig hur assemblerprogrammering går till, men som inte förstått det riktigt ännu. Allra bäst är det givetvis om man har någon expert att fråga. Kombinationen, en expert utrustad med det här programmet, torde vara oslagbar.

Bruksanvisning

Handledningen till programmet är ganska omfattande, ungefär 100 sidor i A5 format, och behandlar inte bara själva programmets funktion. Det finns också ett kapitel som är ägnat åt att beskriva hur man kan undersöka vad som händer och sker i maskinen. Dessutom finns det minneskartor över maskinen, ROM, GROM, 256 bytes RAM, minnesexpansionen under X-BASIC, VDP RAM och CRU adresser. Det mesta av detta kan man finna på andra håll, men sällan samlat på ett ställe. Dessutom ger informationen intryck av att vara mycket omsorgsfullt skriven. Vid det här laget vet jag själv tillräckligt om den här maskinen för att ibland kunna upptäcka hårresande felaktigheter i tekniskt orienterade artiklar. Ännu har jag dock inte hittat något fel i bruksanvisningen till Explorer. Det är ett gott betyg.

Utrustningskrav

Det är det gamla vanliga, var min första tanke. Skivminne, minnesexpansion och X-BASIC eller Mini Memory eller E/A-modulen. Det går också att starta programmet med de laddare som hör till Myarc's och CorComp's diskkontrollkort. Däremot är det inte kompatibelt med Myarc's Micro Expansion System. Priset är USD 24,95 i USA.

Summering

Ett mycket intressant program för alla som vill veta mer om maskinens interna arbetssätt och/eller själva programmerar i assembler. Ofta återfinner man båda egenskaperna i en och samma person. Programmet är faktiskt så intressant att Craig Miller, som skrivit manualen, flitigt uppmanar läsaren att inte hänga framför datorn nätterna igenom. Uppmaningarna är fullt befogade och bygger troligen på hans egen erfarenhet . . .

Liksom för Advanced Diagnostics gäller att programmet är kopieringsskyddat. Du kan alltså inte göra någon säkerhetskopia för eget bruk, beklagligt nog.

Slutligen vill jag inte undanhålla er en upptäckt, eller vad det ska kallas, som jag gjorde med hjälp av Explorer. De som var med på träffen i Staffanstorp i februari vet redan vad jag menar. Genom att ladda in programmet i BASIC eller Extended BASIC kan man se precis vad som händer när dessa interpretatorer arbetar. Explorer har också en instruktionsräknare som håller ordning på hur många assemblerinstruktioner (additioner, jämförelser, hopp etc.) som har utförts. Genom att ge BASIC respektive X-BASIC kommandot

```
PRINT SIN(0.2)
```

och låta Explorer räkna instruktionerna vet jag nu att det behövs drygt 50000 instruktioner för att få ut värdet och sedan lyfta skärmen två rader. Detta gäller för Extended BASIC. Med egna ögon kan man se att det tar lite längre tid i TI BASIC. Mycket riktigt, Explorer räknade till ungefär 125000 (det *ska* vara sex siffror) innan det var klart.

Detta visar, om inget annat, klart vådan av att krångla till en datorarkitektur för mycket. Dock har jag ännu inte upptäckt någon av de förseningsloopar som elaka tungor hävdar finns inbyggda i maskinen. Det hindrar inte att de kan finnas någonstans bland tiotusentals instruktioner . . .

Värt att köpa program från Programbanken?

*Är programmen i föreningens programbank någonting att ha? Är Du rädd för att köpa "grisen i säcken"? **Lennart Thelander** har tagit greppet på frågan och ger sina omdömen om programmen i början på listan.*

01101002
Adressregister
Språk: engelska
Utrustning: ev. skrivare
Omdöme: Ett bra program, väl använda pengar.
Detta är ett adressregister där (nästan) allt kan lagras. Du matar in (i tur och ordning) efternamn, förnamn, c/o adress, gatuadress, postadress, telefonnummer hemma och telefonnummer till jobbet. När du har matat in alla namn får du frågan om du vill spara allt på kassett. Om du svarar 'Y' på frågan kommer allt detta att lagras på kassett. Du kan sedan titta på alla namn, spara allt på kassett, hämta från kassett, ändra i registret, sortera, ta bort, söka namn och skriva ut etiketter på skrivaren. Dessa valmöjligheter väljer du från en menysida. Allt är snyggt uppställt med menyer och frågor. Etiketter kommer med tre i bredd på papperet. Där finns allt utom telefonnumren. Ett minus finns det tyvärr i detta program. Det är att programmet är på engelska (eller egentligen på amerikanska). Det innebär att alla ledtexter är på engelska och att det inte finns några svenska tecken. Det här är ju enkelt att översätta för den som vill.

15103092
Adressregister
Språk: svenska
Utrustning: X-BASIC
Omdöme: Inte lika bra som 01101002E (se ovan).
Det här adressregistret har flera nackdelar jämfört med det ovanstående. Till att börja med finns här bara fyra rader med information: namn, gatuadress, postadress och telefonnummer. Det finns alltså inga möjligheter till c/o adress eller telefonnummer till arbetet. Vidare finns bara en rad med namn. Om man vill sortera på efternamnet hellre än förnamnet måste man mata in SVENSSON, SVEN hellre än SVEN SVENSSON. Det är ju litet dumt. När man matar in registret används ACCEPT med VALIDATE. Det innebär att man alltid måste mata in versaler. Vidare kan man inte mata in kolon som man använder om man t.ex. vill mata in utmaningsredaktörens adress. Han bor på Kämnärsvägen 4:1078. I telefonnummer kan man inte mata in minustecken för att skilja riktnummer från abonnentnummer utan man måste använda det gamla snedstrecket. Man kan heller inte gruppera siffrorna med mellanslag. Vad tycker läsaren är lättast: 042/151873 eller 042 – 151873? En sak till om detta program: de svenska bokstäverna åäö finns under fel tangenter på tangentbordet så man blir tvungen att leta en del innan man hittar dem. Efter litet smågnäll över diverse saker tar vi nu fram storsläggan: Det finns *ingen* möjlighet att få ut någon del av registret på papper. Man måste alltså ha datorn igång för att kunna läsa adresserna. Vidare sparas registret på disk i stället för på kassett. (Det är ju inte alla förunnat att ha disk...)

10121010E
Ordbehandlare
Språk: engelska
Utrustning: 99/4 och Thermal printer
Omdöme: inget att ha för dem som har 99/4A.
Denna ordbehandlare är gjord speciellt för dem som har 99/4 (utan A, den äldsta hemdatorn från Texas). Den datorn hade inte små bokstäver och inte så mycket annat heller. Detta program är fullt av tricks för att få fram små och stora bokstäver på skärm och skrivare (Thermal printer). För den som har 99/4A är detta att gå över ån efter vatten eftersom det redan finns alla tecken på den datorn.

14123039
Ordbehandlare
Språk: Svenska (Svengelska)
Utrustning: (ev. skrivare)
Omdöme: En ganska bra ordbehandlare för den som inte har X-BASIC.
Det här programmet lider av alla brister som uppkommer när man ska göra snygga menyer och tablåer i Basic. Skärmen måste rullas i tid och otid eftersom man bara kan skriva längst ner (med rimlig hastighet). För att det ska gå så snabbt som möjligt har ledtexterna tryckts ihop på så få rader som möjligt och gör ett rörigt intryck. Annars finns här nästan allt man kan begära: Söka text, lägga in rader, ta bort rader, byta ord och flytta om stycken. Tyvärr finns ingen möjlighet att få in kommatecken i strängarna. Jo det finns det ju: man kan ju börja och sluta raden med citationstecken ". Det är bara det att det glömmer man ju oftast med input error som följd. Ledtexter och andra texter skulle må bra av vara på enbart svenska eller enbart engelska. Nu är de flesta på svenska, men en del på engelska, vilket gör ett förvillat intryck.

01121001E
Ordbehandlare.
Språk: engelska
Utrustning: (ev. skrivare)
Omdöme: Ungefär lika bra som föregående ordbehandlare. Det som står om 14123039 (se ovan) gäller även här med följande undantag: Ledtexterna är snygga och prydliga. Tyvärr kan man inte flytta om stycken i texten, men man kan ju lägga till eller ta bort rader var man vill. Vän av ordning ställer sig nu förstås frågan: Vilken ska jag välja? Svaret är att jag inte kan ge något råd. Mitt förslag är att köpa båda och själv ta det bästa ur programmen. Det kostar ju bara 10 eller 15 kronor mer att köpa två än en (beror på om man har disk eller kassett).

12143006
Räntor och annuitet.
Språk: svenska
Utrustning: -
Omdöme: ett mycket prisvärt program.
Detta program beräknar sammansatt ränta (ränta på ränta), sammansatt ränta med årlig insättning och annuitet. Programmet är uppbyggt med prydliga förklarande texter och tydliga anvisningar för inmatning av siffervärden. Programmet kommer snabbt fram till svaret och skriver ut det på skärmen. Därefter kan man köra samma del en gång till eller komma till huvudtablån.

01201001E
Linjär regression
Språk: engelska
Utrustning: -
Omdöme: ett enkelt men bra program.
Detta program begär att man skall mata in ett antal punkter på en rät linje. Programmet räknar då ut den linje som bäst passar till dessa punkter. Man får ut den räta linjens ekvation och får sedan chansen att mata in olika X-värden. Programmet skriver sedan ut motsvarande Y-värde.

Annons

Programming Aids I, kassett, 40:–
Graphing Package, kassett, 40:–
Teach Yourself TI-BASIC, kassett, 40:–
Teach Yourself Extended BASIC, kassett, 40:–
Personal Financial Aids, kassett, 40:–
Mailing List, diskett, 70:–
Electronic Library, diskett 70:–
RS 232, fristående, 700:–
Expansionsbox, hemmabygge, 500:–
(5 slots, fläkt, agg för 2 drivar)
Plus div. program, adm. program

Hans Wickström
Ymergatan 7B
753 25 UPPSALA
tel 018/110418

Från Programbankiren

Föreningen har fått tre freeware-program. *Neatlist* och *Screendump* har vi fått från USA via England där Peter Odelryd har kontakter. Programmen är skrivna av Danny Michael i Alabama. *Disk Manager 99* har kommit från Mike Dodd i Tennessee, USA. De tre programmen är assemblerprogram och kräver diskdrive och expansionsminne. Vilken ytterligare utrustning som krävs nämns i beskrivningen över varje program nedan. Till alla program medföljer även kommenterade källkoder och körinstruktioner på engelska. Programmen distribueras från programbanken för 60:– per program. Varje program fyller i det närmaste en diskett.

Neatlist

Neatlist är skrivet för X-Basic och kräver alltså den modulen. Printer är inte nödvändig utan det går att ange en fil på diskett som utskriftsenhet. Formatet på filen blir då DIS/VAR 254. Det går inte att ange skärmen eller kassett.

För utskriften finns det flera möjligheter. Man kan välja programlistning eller ej och man kan få en variabellista med eller utan referens till de rader där variablerna förekommer. Vidare kan man ange from rad tom rad som man vill ha listade. Om man anger ett för lågt radnummer börjar listningen med det lägsta radnumret i programmet. Motsvarande gäller om man anger ett för högt radnummer.

Man har möjlighet att formatera listningen. Det finns möjlighet att ange radlängd, vänster- och högermarginal, antal rader per sida och antal blankrader högst upp och längst ner på varje sida. Kontrollkoder kan anges för att styra printern tex för att få hoptryckt (condensed) text. Man har möjlighet att ange om kontrollkoderna ska sändas en gång till printern eller om de ska sändas för varje rad.

Alla parametrar som ska matas in har ett defaultvärde men dessa kan ändras med ett medföljande program.

Den bruksanvisning som finns med på disketten är på engelska och är bra. Den talar inte bara om hur de olika parametrarna ska anges utan förklarar också vad som händer om man anger ett felaktigt värde. Vidare berättas om felmeddelanden, hur minnet används av programmet mm.

Om man har flera kommandon på en programrad så delas den programraden upp på flera rader i listningen. Delningen sker efter :: och bara den första raden har radnumret, de andra raderna är indragna. Detta gör att listningen blir lättare att läsa. Några skönhetsfel finns om man jämför med den listning man får med LIST. Radnummer anges med förställda nollor så att man alltid får fem siffror i radnumret. Extra blanktecken läggs till på vissa ställen bla före och efter , (komma) och jämförelsetecken (=). Men som sagt detta är bara skönhetsfel och eftersom källkoden finns med bör den som vill kunna ändra på detta.

Screendump

Screendump finns i två versioner. En för Editor/Assembler eller Mini Memory och en för X-Basic. Det behövs alltså någon av dessa moduler förutom printer med bit map grafik. Programmet har testats på Epson MX80 och Gemini 10X. Jag själv har kör det på en Compu Mate CP80 typ II och där fungerade det utan problem. Det bör fungera med alla printrar som har samma kontrollkoder för grafik och samma bit image grafik som ovanstående. I källkoden är de avsnitt som måste ändras för andra printrar tydligt markerade. Screendump kan bara användas för att dumpa en skärm som är i Graphics Mode.

När man ska dumpa en skärm kan man ange fyra olika parametrar. Den första är antal blanktecken från vänstermarginalen. Om man inte anger antal blanktecken kommer programmet att centrera bilden på pappret. Det finns möjlighet att rotera bilden 90 grader medurs vid dumpningen. Det anges med R i CALL LINK kommandot. Vidare finns det möjlighet att dumpa skärmen till dubbel storlek på pappret. Det sker genom D i CALL LINK, då får man automatiskt en vridning 90 grader medurs för att bilden inte ska komma utanför papperskanten. Den sista möjligheten man har är att få bilden negativ, dvs figuren är vit mot en svart bakgrund på pappret. Hur man anger de olika parametrarna framgår tydligt av bruksanvisningen.

Av bruksanvisningen, som är på engelska, framgår också hur man använder programmet tillsammans med en "load interrupt switch". Det är en hårdvaruändring av datorn som har beskrivits bla i *Enthusiast '99* magazine.

Man kan även dumpa skärmen från tex spelmoduler. Där finns det vissa krav, bla att bilden måste vara i graphics mode. Parsec tex går inte att dumpa eftersom den skärmen är i Bit Map Mode. Vidare måste screen image table börja på adress >0800. Hur man gör omtalas i bruksanvisningen, som är lika bra som den för Neatlist. Det är ju samma person som har skrivit båda programmen.

För bruksanvisningen, en DIS/VAR 80-fil, finns ett X-Basicprogram som skriver ut den på skärmen. Eftersom bruksanvisningen finns på en DIS/VAR 80-fil bör det inte vara något problem att skriva ett Basic-program som listar den.

Disk Manager 99

Disk Manager 99 finns i två versioner. En som kan köras från Basic med Editor/Assembler eller Mini Memory. Den andra versionen är för X-Basic. Någon av dessa moduler behövs alltså.

Man kan initialisera disketter antingen enkel- eller dubbelsidigt och antingen med enkel densitet eller dubbel (förutsatt att man har rätt diskkontrollkort förstås – *Red.*). En katalog över disketter går att få antingen på skärmen eller på en printer. Det går att skydda och ta bort skyddet på filer. Vidare kan man byta namn på filer och disketter samt testa disketter. Allt detta sker med hjälp av CALL LINK antingen från tangentbordet eller i ett program. Det man inte kan göra är att kopiera filer eller hela disketter, men det gör väl inte så mycket. Den stora fördelen med det här programmet är att man kan ha det i minnet hela tiden och man behöver därför inte slå ifrån datorn och byta modul när man behöver tillgång till en Disk Managerfunktion.

Bruksanvisningen, som är på engelska, är inte lika fullständig som för Neatlist och Screendump men den förklarar tillräckligt för att det inte ska vara några problem vid användningen.

Varning för lycksökare!

I kölvattnet på olika verksamheter som har en viss framgång brukar det ofta dyka upp mindre seriösa personer som är ute efter att tjäna snabba pengar. De flesta av dessa har för länge sedan slutat att agera runt TI-99/4A – de flydde som råttor från det sjunkande skeppet när TI slutade tillverkningen.

Men det finns en del kvar. En av dessa verkar vara en firma som kallar sig *National Assistance Group*, med adress i Florida. De utger sig för att vara en användargrupp för TI-99, men det verkar i själva verket vara en vanlig postorderfirma. De påstår sig ha tiotusentals medlemmar över hela världen och deras annonser är ofta undertecknade av personer som titulerar sig "doktor" odyl. Program som de *säljer* dyrt inkluderar freeware av den typ som normalt distribueras mot kostnader för diskett och porto (tex av vår förening).

I ett svar på ett brev från Jan Alexandersson har de påstått att "många nya medlemmar från Sverige har gått med det senaste året". Vi utgår från att detta är lika lögnaktigt som deras påstående om 35000 medlemmar över hela världen. Att vi överhuvudtaget nämner deras existens beror på att de annonserat en del på sistone i tidningen *Compute*, en tidning som numera knappast innehåller något material för TI-99, men som distribueras över hela Sverige och därför kanske läses av en del Texas-ägare. Skulle någon ha gått med i denna suspekta grupp är vi tacksamma för upplysningar om vilket material de eventuellt skickat ut.

FORTH spalten

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

Den inre interpretatorn

En högnivådefinition i FORTH – dvs ett ord definierat med kolon – är en lista med adresser till andra FORTH-ord. Varje sådan adress är en cfa – kodfältsadress – i ett ord som skall utföras. Varje cfa pekar på en cell – kodfältet – som innehåller en adress till körbar kod.

I ett kolon-ord ligger de här adresserna uppradade i rätt ordning i parameterfältet. Jag tänker nu titta litet närmare på hur FORTH hanterar adresserna och nystar upp härvan. Just detta är grundläggande för FORTH som språk och gäller för alla FORTH-tillämpningar. De tillämpningar som är aktuella för oss – TI 3.0, PB 2.0 och 2.1, FORTH&99 – är dessutom lika även i det maskinspecifika såsom minnesadresser, maskinregisteranvändning m m.

Låt oss börja med att se på den virtuella maskin som FORTH bygger på. Det är en tänkt maskin som inte finns konkret utan som simuleras av det program som implementerar FORTH på olika konkreta maskiner. Jag beskriver bara några viktiga drag i den virtuella maskinen. Att det finns en virtuell maskin är förresten en viktig faktor bakom FORTHs portabilitet, dvs möjligheten att köra samma källkod på olika maskiner. Några centrala register i den virtuella maskinen är

- SP Data Stack Pointer
- R Return Stack Pointer
- IP Interpretive Pointer
- W Current Word Pointer

SP och R pekar på toppen av resp stack och ökas resp minskas när man lägger något på stacken (eng.: push) eller lyfter något av stacken (eng.: pop).

IP pekar på en adress inuti en högnivådefinition och är – håll i Dig nu – en pekare till en pekare till en pekare till körbar kod.

W pekar mest på den cfa vars kod utförs. I ett visst skede av interpreteringen pekar W på pfa för det FORTH-ord som skall utföras.

De här pekarna utnyttjas av för FORTH standardiserade rutiner för interpreteringen och styr på så sätt körningen. En central rutin för detta är den som kallas den inre interpretatorn. Med en inre interpretator i FORTH menas:

> en uppsättning procedurer, vanligen programmerade i maskinkod, som utför FORTH-ord genom att läsa och tolka information i ordens parameterfält. Adressen till de berörda procedurerna finns lagrade i det aktuella FORTH-ordets kodfält. FORTH-definitioner av samma klass använder samma procedurer, dvs de har samma adress i kodfältet.
>
> Den mest centrala interpretatorn är den som hör till kolondefinitionerna. Den kallas

- DOCOL Adress Interpreter

Vidare finns där

- DOCON Constant Interpreter
- DOVAR Variable Interpreter
- DOUSE User Variable Interpreter
- .+2 Code Interpreter

Den sista, som alltså hör till de ord som definieras med CODE – de som bär den körbara koden i sitt eget parameterfält – kan tyckas besynnerlig. Det kan ju klarna när man tänker på att man kommer från kodfältet i ett ord till parameterfältet just genom att göra +2 till adressen i de flesta FORTH-varianter.

Parentetiskt: när man med <BUILDS .. DOES> bildar nya definierande ord skapar man just en ny inre interpretator för denna ord-klass, skriven i högnivå-FORTH och placerad efter DOES>.

Den interpretator vi skall intressera oss närmare för är DOCOL. I TI 3.0 och de andra versioner som nämnts kännetecknas alla kolon-definitioner av att i deras kodfält finns adressen >8334. På den adressen startar den rutin som kallas DOCOL. Det är inte ett vanligt FORTH-ord utan ett maskinkodsprogram. Adressen ligger i det utrymme som i TI99/4A kallas scratch-pad – 256 bytes CPU-RAM på adresserna >8300 – >83FF. Detta är det enda CPU-RAM som finns i 99-ans konsol, som emellertid innehåller en del VDP-RAM. Scratch-pad ligger i direkt anslutning till 99-ans 16-bitars data-buss och är det snabbaste RAM som finns i maskinen. Därför används det till de mest frekventa bitarna av FORTH:

- >8300–>831F Registerarean (R0/R15)
- >832E–>8347 DOCOL m fl liknande

Innehållet i det sista blocket ovan är smart kodat med ett effektivt utnyttjande av kodsekvenserna för flera rutiner. I källkodsform ser det ut så här:

```
>832E DODOES DECT SP
>8330        MOV  W,*SP     * pfa till stacken
>8332        MOV  LINK,W
>8334 DOCOL  DECT R
>8336        MOV  IP,*R     * IP->R
>8338        MOV  W,IP
>833A $NEXT  MOV  *IP+,W
>833C DOEXEC MOV  *W+,TEMP1
>833E        B    *TEMP1
>8340 $SEMIS MOV  *R+,IP    * R->IP
>8342        MOV  *IP+,W
>8344        MOV  *W+,TEMP1
>8346        B    *TEMP1
```

De register som skall finnas i den virtuella maskinen är implementerade i 99-an med följande equates

- SP (=R9) pekare t. parameterstacken
- R (=R14) pekare t. returstack
- IP (=R13) instruktions-pekare
- W (=R10) arbetsregister

Dessutom används ytterligare några register av intresse här:

- TEMP1 (=R1) temporärt register
- LINK (=R11) återhopp vid BL
- NEXT (=R15) pekare t $NEXT

NEXT är en viktig rutin i FORTH som används som avslutning på alla primitiver bl a. Den avslutar ju också DOCOL. Ibland kallas NEXT för den inre interpretatorn – ett språkbruk som jag finner litet förvirrande.

Vi ser ganska lätt att kodsekvensen ovan innehåller två skilda rutiner. En slutar på >833E och den andra, benämnd $SEMIS, på >8346. Den som slutar på >833E har flera ingångar, benämnda DODOES, DOCOL, $NEXT och DOEXEC. DODOES är interpretator för DOES> och DOCOL för kolon. $NEXT är den körbara koden för NEXT, som adresseras via R15. DOEXEC slutligen är interpretator för ett enda ord som särbehandlas, nämligen EXECUTE. Det är det första ordet i ordlistan också. $SEMIS är interpretator för ;S, som ju avslutar alla kolondefinitioner.

Låt oss börja med att titta på DOCOL. När DOCOL exekverar händer följande:

1. Returadressen till nästa ord lagras i returstacken;
2. IP sättes till ordets pfa plus 2;
3. W sättes till pfa för den första rutinen som ingår i ordet (på adress pfa);
4. Hopp till adressen i första rutinens kodfält (dvs ordet på pfa i det anropade ordet).

När DODOES exekverar händer följande:

1. Pfa för det av <BUILDS skapade ordet lagras på parameterstacken;
2. En "simulerad" pfa lagras i W (pekar på första instruktionen efter DOES>);
3. DOCOL exekveras.

När $SEMIS exekveras händer följande:

1. Översta talet på returstacken ökas med 2 och poppas till IP;
2. Nästa instruktions pfa lagras i W;
3. Hopp till adressen i nästa ords kodfält.

När en kolondefinition skall exekveras lagras adressen till nästa ord på returstacken (av DOCOL). När sedan ;S exekveras sist i raden poppas returstacken (av $SEMIS) och exekveringen fortsätter på nästa adress. Dvs förutsatt att Du inte manipulerat med returstacken på ett olämpligt sätt! Har Du manipulerat fel låser sig kanske processorn (den försöker exekvera koden på en slaskadress). Har Du manipulerat på ett meningsfullt sätt (se tex (.") i FORTH-spalten 4-85) styrs programflödet om på det sätt Du önskar (tex för att ge plats för data i parameterfältet).

Returstacken

Detta för oss över till returstacken och dess möjligheter. Den kan även användas för att tillfälligt lagra data som i ?PRIME. ?PRIME testar om u, som skall vara ett heltal>0, är ett primtal. f är då sant, eljest falskt:

```
?PRIME ( u -- f ) >R 2
 BEGIN DUP R 0 ROT U/ DROP
 WHILE 1+
 REPEAT R> = ;
```

?PRIME lagrar först värdet på datastacken i returstacken med >R. Detta värde kan sedan kopieras till datastacken med ordet R. Sista gången kopierar Du inte värdet med R utan flyttar det med R> för att poppa returstacken innan ;.

Ett annat sätt att lagra data i returstacken visades i Don Taylors MACRO: i Programbiten 4-85. Kommandot LOAD använder också returstacken för att lagra data.

I poly-FORTH (dialekten i "Starting FORTH!") finns ett ord som heter I', vilket inte finns i fig-FORTH:

I kopierar översta elementet på returstacken till datastacken (samma som R).
I' kopierar näst översta talet.
J kopierar näst näst översta talet.

Det finns ett lite klurigt sätt att definiera I' i fig-FORTH:

```
: I' J ;   ( heureka!!)
```

Detta betyder inte att I' blir en kopia av J. När I' anropas läggs istället ytterligare en adress på returstacken. Simsalabim! Det som innan var I' blir J!

Du kan ju fundera över om det här sättet att definiera I' är helt OK eller om det innehåller några problem som kan behöva rättas till.

FORTHs grunder

Tycker Du att denna och de senaste FORTH-spalterna har varit svåra att förstå? Misströsta inte! Bättring utlovas med lite mer elementära aspekter på FORTH.

Avsikten med de tre-fyra senaste FORTH-spalterna har varit att gå igenom mer avancerade detaljer och fästa något om dessa på papper för framtida bruk. Om Du känner Dig som en nybörjare och har haft svårt att följa med råder jag Dig: läs de här spalterna på nytt efter en tid! Det är förunderligt vad saker och ting klarnar efter ett tag. Och skulle det inte klarna – skriv gärna så får vi resonera om Dina frågor. Den här spalten kan också fungera som frågespalt.

I kommande spalter planerar jag att ta upp bl a PB-FORTH, heltalsaritmetik, liststrukturer, en AI-parser för äventyrsspel, grafik, interruptrutiner, simulering av fysikaliska processer samt en del annat.

Om Du har några förslag tar jag gärna emot dem.

För säkerhets skull: jag kommer aldrig försöka mig på någon grundkurs i FORTH. En bra sådan får Du via *"Starting FORTH!"* av Leo Brodie. Jag har ingen önskan att överträffa honom. Det är nog så att jag inte ens skulle kunna. Möjligen – det finns en del önskemål om detta – kan vi kanske ordna någon form av cirkelverksamhet? Jag vill gärna veta vad Du har för synpunkter på den saken!

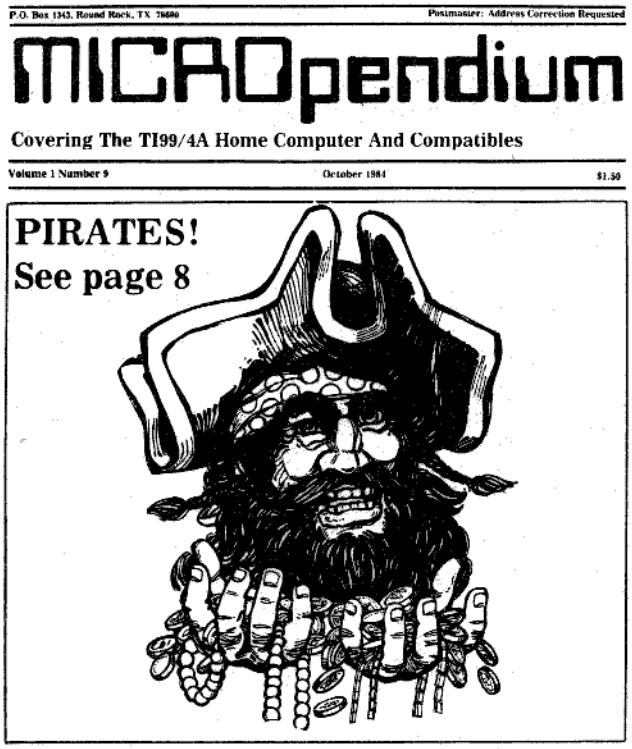
Till den speciella intressegruppen för FORTH har nu anslutit sig drygt 30 medlemmar. Fem tjocka utskick har distribuerats av Peter Odelryd (c:a 5 kr/st!).

Vi går framåt, sakta men säkert!

Go FORTH, easy ahead!

MICROpendium – nyhetstidning för 99-an

Vi har ofta publicerat "skvaller" och nyheter i den här tidningens spalter. Den främsta nyhetskällan har varit en amerikansk tidning som heter *MICROpendium*. Tidningen kommer ut punktligt en gång i månaden och har just kommit in på sin tredje årgång. Den är oftast på 48 sidor (varav knappt hälften annonser) och är framställd på billigt papper utan färgtryck. En stor del av innehållet brukar ägnas åt recensioner av program- och maskinvara. För dem som vill veta vad som händer på 99-fronten i USA är tidningen oumbärlig. Programlistningar (oftast av det kortare slaget) och tips från läsarna finns det också som brukar vara matnyttiga. En årsprenumeration kostar 35 dollar med flyg och 23,50 dollar med ytpost. Tidningens adress är: MICROpendium, P.O. Box 1343, Round Rock, TX78680, USA.



Annons

TI 99/4A grundenhet med tillbehör säljes
Expansionsbox, 32 Kb minnesexp, RS232-interface, Diskdrive (intern), Speech synthesizer, Editor/assembler, Mini Memory, Terminal Emulator, FORTH (föreningens och Texas), manualer och instruktioner (bl a 9900 Family System Design och The 9900 Family Data Book).
Pris 6000 kr för allt.

Anders Bohman
Fornhöjdsvägen 23
15158 Södertälje
tel. 0755/64229.

TI 59
Vertikalkurvor, konkava och konvexa

Från Jan Nilsson har vi fått in följande program för utsättning av vertikalkurvor inom vägbyggnad för TI-59 med skrivare.

Vertikalkurvor, konvexa och konkava.

Datalagring

```
        R ==> STO 00 (-R vid konkav)
        a ==> STO 01
        b ==> STO 02
 höjd   C ==> STO 03
```

detta är vad som måste in före programkörning.

Sedan sker Automatisk datalagring

L_1==> RCL 05 (avstånd TP_A -> D)

L_2==> RCL 06 (avstånd TP_B -> D)

L_{Tot}==> RCL 07 (korda horisontellt)

$L_{Tot}/2$ ==> RCL 08 (avstånd $TP_{A,B}$ -> C horisontellt)

Y_1 ==> RCL 09 (höjdskillnad TP_A -> D)

Y_2 ==> RCL 10 (höjdskillnad TP_B -> D)

Programkörning
TRYCK A => vilket ger utskrift:
Höjd TP_A, Höjd TP_B, Höjd D.

Mata in steglängd från punkten D,
TRYCK B => vilket ger utskrift Höjd Y.

Beräkningsexempel:

```
  Radie   2500 m
  a         20 p/mille
  b         30 p/mille
  höjd     +40 m

steglängd för TPA: 50 m
steglängd för TPB: 75 m

som ger föjande svar:
          L1     50
          L2     75
          Ltot  125
          L/2    62.5
          Y1      0.5
          Y2      1.12
         ------------
          TPA + 38.75
          TPB + 38.125
          D     39.25
```

```
 2500.        00
   20.        01
   30.        02
   40.        03
    0.        04
   50.        05
   75.        06
  125.        07
 62.5         08
  0.5         09
1.125         10
38.75         11
38.125        12
39.25         13
   75.        14
38.750      +TPA
38.125      +TPB
39.250      +D
50.000
38.750      +Y
75.000
38.125      +Y
```

Exemplet hämtat ur:
Planmätning och avvägning
av Sven Dalstrand
Liber förlag

sid 474.

C = Brytpunkt

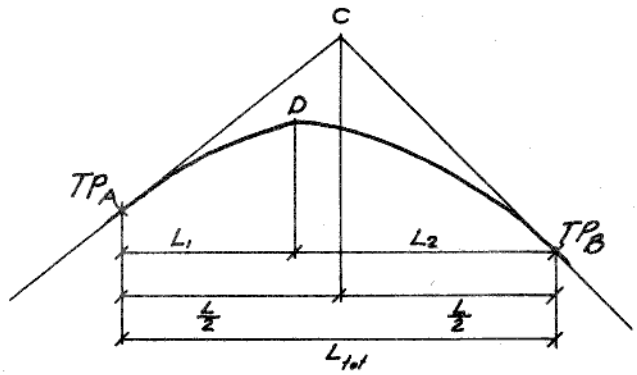
a = Tangentlutning vid TP_A

b = Tangentlutning vid TP_B

1. $L_1 = R * \frac{a}{1000}$
2. $L_2 = R * \frac{b}{1000}$
3. $L_{Tot} = L_1 + L_2$
4. $\frac{L_{Tot}}{2} = L_{TP}$ -> C

Förutsättningar:

Radien	= känd
Tangentlutningen	= känd
Brytpunktssektionen	= känd
Brytpunktshöjd	= känd



5. $Y_1 = \frac{L_1 * L_1}{2 * R}$ Y_1 = krönpunktens höjd över TP_A
6. $Y_2 = \frac{L_2 * L_2}{2 * R}$ Y_2 = krönpunktens höjd över TP_B
7. höjd TP_A = höjd C - $\frac{\frac{L}{2} * a}{1000}$
8. höjd TP_B = höjd C - $\frac{\frac{L}{2} * b}{1000}$
9. höjd D = höjd TP_A + Y_1 eller höjd TP_B + Y_2
10. höjd Y = höjd D - $\frac{L}{2} * \frac{L}{R}$

Y = Valfri detaljpunkt på kurvan.

L = Längd från D

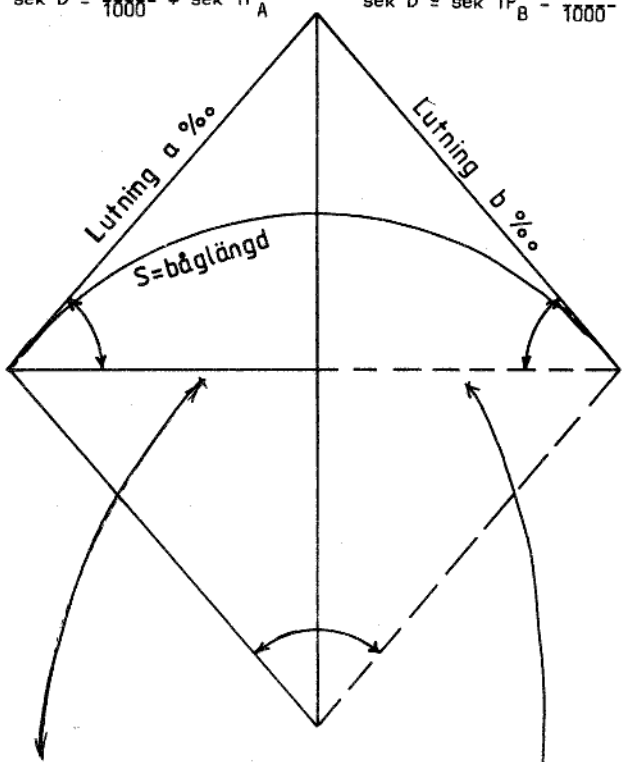
TI 59

$\sin v_1 \approx \tan v_1 = \frac{a}{1000}$ $\qquad$ $\sin v_2 \approx \tan v_2 = \frac{b}{1000}$

(vid små vinklar)

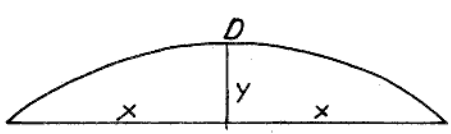
$S = \frac{(a+b) \times R}{1000}$

sek D = $\frac{a \times R}{1000}$ + sek TP_A $\qquad$ sek D = sek TP_B - $\frac{b \times R}{1000}$



$L_1 = R * \sin v_1 - R * \tan v_1 = R * \frac{a}{1000}$

$L_2 = R * \sin v_2 - R * \tan v_2 = R * \frac{b}{1000}$



Kordasatsen: $X^2 = Y(2R-Y)$

$X^2 = 2R * Y - Y^2$

$X^2 \approx 2R * Y$ eller $Y \approx \frac{X^2}{2 * R}$

(Då Y^2 är litet i förhållande till R kan Y^2 försummas)

$Y_1 = \frac{L_1^2}{2 * R}$ och $Y_2 = \frac{L_2^2}{2 * R}$

```
000  76 LBL     065  43 RCL     129  42 STO
001  11  A      066  03  03     130  12  12
002  80 GRD     067  75  -      131  69 OP
003  58 FIX     068  53  (      132  06  06
004  03  03     069  43 RCL     133  22 INV
005  43 RCL     070  08  08     134  58 FIX
006  00  00     071  65  ×      135  04  4
007  65  ×      072  43 RCL     136  07  7
008  43 RCL     073  01  01     137  01  1
009  01  01     074  55  ÷      138  06  6
010  55  ÷      075  01  1      139  08  8
011  01  1      076  00  0      140  00  0
012  00  0      077  00  0      141  08  8
013  00  0      078  00  0      142  00  0
014  00  0      079  95  =      143  69 OP
015  95  =      080  42 STO     144  04  04
016  42 STO     081  11  11     145  58 FIX
017  05  05     082  22 INV     146  03  03
018  43 RCL     083  58 FIX     147  43 RCL
019  00  00     084  04  4      148  11  11
020  65  ×      085  07  7      149  85  +
021  43 RCL     086  03  3      150  43 RCL
022  02  02     087  07  7      151  09  09
023  55  ÷      088  03  3      152  95  =
024  01  1      089  03  3      153  42 STO
025  00  0      090  01  1      154  13  13
026  00  0      091  03  3      155  69 OP
027  00  0      092  69 OP     156  06  06
028  95  =      093  04  04     157  91 R/S
029  42 STO     094  58 FIX     158  76 LBL
030  06  06     095  03  03     159  12  B
031  85  +      096  43 RCL     160  42 STO
032  43 RCL     097  11  11     161  14  14
033  05  05     098  69 OP     162  99 PRT
034  95  =      099  06  06     163  22 INV
035  42 STO     100  22 INV     164  58 FIX
036  07  07     101  58 FIX     165  04  4
037  55  ÷      102  04  4      166  07  7
038  02  2      103  07  7      167  04  4
039  95  =      104  03  3      168  05  5
040  42 STO     105  07  7      169  08  8
041  08  08     106  03  3      170  00  0
042  43 RCL     107  03  3      171  08  8
043  05  05     108  01  1      172  00  0
044  33 X²      109  04  4      173  69 OP
045  55  ÷      110  69 OP     174  04  04
046  53  (      111  04  04     175  58 FIX
047  02  2      112  58 FIX     176  03  03
048  65  ×      113  03  03     177  43 RCL
049  43 RCL     114  43 RCL     178  13  13
050  00  00     115  03  03     179  75  -
051  95  =      116  75  -      180  53  (
052  42 STO     117  53  (      181  43 RCL
053  09  09     118  43 RCL     182  14  14
054  43 RCL     119  08  08     183  33 X²
055  06  06     120  65  ×      184  55  ÷
056  33 X²      121  43 RCL     185  02  2
057  55  ÷      122  02  02     186  55  ÷
058  02  2      123  55  ÷      187  43 RCL
059  55  ÷      124  01  1      188  00  00
060  43 RCL     125  00  0      189  95  =
061  00  00     126  00  0      190  69 OP
062  95  =      127  00  0      191  06  06
063  42 STO     128  95  =      192  91 R/S
064  10  10
```

Programspråket C för 99:an

av Björn Gustavsson

Ännu ett programspråk kan nu användas på 99:an, nämligen C. Kompilatorn, som kallas för c99, har anpassats för 99:an av Clint Pulley, Ontario, Canada. Han erbjuder c99 som ett användarstött program, vilket innebär att det får distribueras fritt om man anger källan och endast tar ut en rimlig distributionsavgift och ersättning för disketten. Användare som tycker om programmet kan sedan sända pengar till programförfattaren.

Språket C skapades i början av 70-talet av Dennis Ritchie. Namnet C kommer från två språk som var föregångare till C: BCPL och B. Det första större programmet som skrevs i C var UNIX, ett operativsystem ursprungligen för minidatorer, men numera tillgängligt även för mikrodatorer. Även de flesta hjälpprogram och tillämpningsprogram, även själva C-kompilatorn, är skrivna i C.

Under de senaste åren har C blivit ett väldigt populärt programspråk för persondatorer, t ex IBM PC, men även för CP/M-baserade datorer och datorer med 68000. Stora programvaruföretag som Microsoft, Ashton-Tate, Micro-Pro mfl använder C.

Varför detta stora intresse för C? Vilka fördelar har C jämfört med andra språk?

Vi kan först jämföra med assembler som bara för 5 år ofta användes för att skriva applikationsprogram för mikrodatorer. Anledningen var att det var enda möjligheten att få tillräckligt snabba och/eller kompakta program. Nackdelen är att det är tidsödande att skriva program i assembler; dessutom är källkoden svår att förstå och ändra, och koden kan inte flyttas till en annan dator med annan processor.

När man skriver i ett högnivåspråk måste programmet först översättas, kompileras, till assemblerkod eller maskinkod för att kunna köras. Maskinkoden som blir resultatet blir sällan lika bra som om programmet hade skrivits direkt i assembler; programmet blir alltså större och långsammare. Med dagens sjunkande priser på minnen och allt snabbare processorer blir det mindre och mindre intressant. Man bekymrar sig mer om kostnaden för att utveckla programmen, dvs hur länge programmeraren håller på för att skriva ett program. Olika undersökningar har visat att en programmerare blir upp till 10 gånger mer produktiv genom att skriva i ett lämpligt högnivåspråk.

Vad beträffar exekveringshastighet går det för det mesta att få program skrivna i högnivåspråk tillräckligt snabba. Man brukar säga att 90 procent av tiden används för att utföra 10 procent av programmet. Med ett program som brukar kallas "profiler" kan man mäta var i programmet största delen av tiden förbrukas. Genom att skriva om endast den delen av programmet i assembler blir hela programmet snabbare. Ofta finner man att masslagringsenheter, t ex diskettstationer, är så långsamma att det inte spelar någon roll hur mycket man snabbar upp programmet.

C är ett högnivåspråk på en ganska "låg" nivå. Det innebär att alla satser som finns i språket lätt kan översättas till maskinkod för de flesta typer av datorer. Det finns inga satser för att direkt hantera sammansatta datatyper, som tex strängar. Om man vill jämföra två strängar måste man använda en loop och jämföra tecken för tecken. Det finns heller inga inbyggda satser för in- och utmatning och filhantering. Däremot finns de flesta operatorer som finns i assembler, och som normalt saknas i andra högnivåspråk: bl a AND, OR, XOR och skiftinstruktioner.

Detta innebär att man sällan har något behov att använda assembler. Man kan göra det mesta i C och C-koden blir ofta tillräcklig effektiv.

Till varje C-kompilator brukar det finnas ett bibliotek med färdigskrivna funktioner för filhantering, in- och utmatning, stränghantering etc. Det finns en standard för hur ett sådant bibliotek skall se ut, men om man inte gillar standardfunktionerna kan man skriva helt egna funktioner för all in/ut-hantering. Standardbiblioteket brukar skrivas huvudsakligen i C, med vissa inslag av assembler.

C tillhandahåller de kontrollstrukturer som behövs för att skriva strukturerade program: IF-sats (if), olika sorters loopar (while, for och do) och en CASE-sats (switch). En GOTO-sats (goto) finns också, men det är sällan man har någon användning för den.

Om man skall jämföra C med andra språk så kan man säga att C hör till Algol-familjen, dit även Pascal hör. Jämfört med Pascal uttrycker man sig betydligt mer "kort och koncist" i C. I Pascal används t ex de reserverade orden BEGIN och END för att dela in satser i en grupp; i C används klammerparenteser. I C finns kortformer för att öka eller minska en variabel med 1. Jämför följande utdrag från ett Pascal-program:

```
a := foo[i];
i := i + 1;
j := j + 1;
b := bar[j];
```

med motsvarande kod i C:

```
a = foo[i++];
b = bar[++j];
```

Kritiker av C brukar slå ner på att C-program är kryptiska och svårlästa. Det är de också för den som inte kan C utan endast språk som t ex Pascal eller BASIC. Men den som inte kan stenografi förstår heller ingenting av ett stenogram, medan den som kan det både snabbt läser och skriver stenogram. C kräver lite mer ansträngning att komma in i än t ex Pascal.

Visst kan man missbruka C och skriva program som ingen begriper – men det kan man i alla språk. Rätt utnyttjat kan man skriva program som är lättare att förstå just för att de innehåller mindre kod än motsvarande program i andra språk.

C-kompilatorn för 99:an

Kompilatorn är anpassad från Small-C, skriven av Ron Cain, som publicerades i Dr Dobb's Journal nr 45, 1980. Small-C har blivit en klassiker och många har förbättrat den ursprungliga kompilatorn eller anpassat den till nya datorer eller operativsystem. Att anpassa själva kompilatorn är relativt enkelt eftersom hela kompilatorn är skriven i C; man behöver bara ändra den del av kompilatorn som genererar assembler-kod för en viss processor. (Den ursprungliga kompilatorn genererade kod för 8080.)

Biblioteksfunktionerna är delvis skrivna i assembler och dessa är givetvis beroende på den dator/operativsystem som man använder.

För att kunna använda C-kompilatorn på 99:an behöver man Editor/assembler. Man gör på följande sätt.

1. Man matar in källkoden för C-programmet med editorn och sparar den på en fil. Observera att alla reserverade ord i C endast får skrivas med små bokstäver (dvs små versaler på 99:an).
2. Man startar kompilatorn och anger namnet på källkodsfilen och svarar på några fler frågor. Kompilatorn skapar en fil med assembler-källkod.
3. Om man har lust kan man köra ett program som förbättrar det genererade assemblerprogrammet så att det förhoppningsvis blir något snabbare.
4. Man startar assemblern och assemblerar källkoden som kompilatorn genererade.
5. Man kör programmet.

Språket

Kompilatorn stöder en delmängd av C. Denna delmängd är tillräcklig för att kompilatorn själv skall kunna kompileras. Här följer kort beskrivning av språket.

Det finns endast två grundläggande datatyper:

1. char – 8-bitars heltal (byte)
2. int – 16-bitars heltal (word)

Man kan deklarera vektorer ("arrays") bestående av dessa datatyper och pekare till datatyperna. Däremot kan man inte deklarera vektorer av pekare. Exempel:

```
char bar;   /* 8-bitars heltal */
int foo;   /* 16-bitars heltal */
char text[80];   /* vektor med plats för 80 bytes (tecken) */
int number[100]; /* 100 heltal */
char *name;   /* pekare till ett eller flera bytes */
char *name[50];   * ej tillåtet!! (vektor med pekare till bytes) */
```

Det finns två typer av variabler – globala (åtkomliga av alla funktioner) och lokala (endast åtkomliga från en funktion).

Det finns tre typer av konstanter.
1. Ett decimalt tal.
2. En eller två tecken inom enkla citationstecken. *Exempel:*
'A' (motsvarar 65 = ASCII-koden för A)
'AB' (förbjudet i de flesta andra C-kompilatorer)
3. Noll eller flera tecken inom dubbla citationstecken. Värdet är en pekare till strängen som kompilatorn automatiskt lagrar i minnet. *Exempel:* "Jag är en sträng"

I uttryck får följande unära operatorer användas.
1. "–" Negerar uttrycket. *Exempel:* –2
2. "*" Hänvisar till elementet som uttrycket pekar på (uttrycket måste vara ett pekarvärde) *Exempel:* *name
3. "&" Ger adressen till uttrycket, om det har en adress. *Exempel:* &foo
&1000 (*ej tillåtet*, har ingen adress)
4. "++" Ökar uttrycket med 1. Om det används på en heltalspekare ökas pekaren med 2 för att peka på nästa heltal. Se exempel ovan i texten. Endast tillåtet på uttryck som har en adress. *Exempel:*
v[i++] (ökar i efter att värdet har använts)
v[++i] (ökar i innan värdet har använts)
1000++ *(ej tillåtet)*
5. "--" Minskar uttrycket med 1. Fungerar i övrigt som "++".

Följande binära operatorer finns.
1. "+" Lägg ihop uttrycken. *Exempel:* foo + 50
2. "–" Subtraktion.
3. "*" Multiplikation.
4. "/" Heltalsdivision.
5. "%" Modulo. Ger resten när det första uttrycket delas med det andra. *Exempel:* 5 % 3 ger 2
6. "|" Bitvis OR.
7. "^" Bitvis XOR.
8. "&" Bitvis AND.
9. "=" Tilldelning. Görs från höger till vänster.
Exempel: bar = foo + 50
x = y = z = 0;

Följande logiska operatorer finns.
1. "==" Test för likhet.
2. "!=" Test för icke likhet.
3. "<" Test för mindre än.
4. ">" Test för större än.
5. "<=" Test för mindre än eller lika med.
6. ">=" Test för större än eller lika med.

Följande satser finns.
1. uttryck;
Exempel: i++;
i = 0;
2. if (uttryck) sats
Utför satsen om uttrycket är skilt från 0.
Exempel: if (foo == 5) bar = 6;
3. if (uttryck) sats1 else sats2
4. while (uttryck) sats
Utför satsen medan uttrycket är skilt från 0.
5. break; Avbryt den innersta while-loopen.
6. continue; Starta om en while-loop från början.
7. return;
Återvänd från en funktion utan att returnera något värde.
8. return uttryck;
Återvänd från en funktion och returnera ett värde.
9. ; Tom sats.
10. { sats; ... sats; }
Satsgruppering. Alla satser inom klammer-parentesen räknas som enkel sats.

Det finns en enkel för-processor med två typer av direktiv.
1. #include filnamn
Den angivna filen inkluderas som källkod. *Anmärkning:* I standard-C måste denna sats skrivas: #include "filnamn" eller #include <filnamn>
Exempel (c99): #include "DSK1.STDIO"
2. #define namn sträng
Namnet som anges kommer att bytas ut mot den angivna strängen på alla ställen namnet förekommer i programmet.
Exempel: #define TABSIZE 1000
#define MAXLENGTH 80
3. #asm ... #endasm
Allting mellan dessa två direktiv antas vara assemblerkällkod och kopieras oförändrat till filen med assemblerkällkod som kompilatorn genererar.
Anmärkning: Dessa direktiv finns ej i standard-C.

Biblioteksfunktionerna

De tillgängliga funktionerna beskrivs kortfattat nedan. De funktioner som absolut inte finns i ett vanligt standardbibliotek har markerats. Filfunktionerna verkar vara kompatibla med standardfunktionerna, förutom att unit måste deklareras annorlunda. (int unit i stället för FILE *unit)
Argumenten har följande betydelser.

```
int c, key;        /* Heltal för att lagra ett tecken */
char *string;      /* Pekare till sträng */
char *buf;         /* Buffert för att lagra flera tecken */
int status;        /* Statuskod */
int rad, kol;      /* Rad och kolumn */
int unit;          /* Enhetsnummer för fil. ej standard */
char *mode;        /* Sträng som anger hur filen skall användas */
int error;         /* Felkod */
int max;           /* Maximum antal tecken att läsa */
```

Funktionerna är följande.

```
c = getchar();          /* Läs ett tecken från tangentbordet */
putchar(c);             /* Skriv ut ett tecken på skärmen */
buf = gets(buf);        /* Läs en hel rad från tangentbordet */
puts(string);           /* Skriv ut strängen på skärmen */
exit(status);           /* Avbryt programmet */
abort(status);          /* Avbryt programmet. ej standard */
locate(rad, kol);       /* Flytta markören till rad, kolumn. ej standard */
key = poll(c);          /* Känn av tangentbord. ej standard */
unit = fopen(name, mode);   /* Öppna en fil */
error = fclose(unit);       /* Stäng en fil */
c = getc(unit);             /* Läs ett tecken från en fil */
error = putc(c, unit);      /* Skriv ett tecken till en fil */
buf = fgets(buff, max, unit); /* Läs en rad från en fil */
fputs(string, unit);        /* Skriv en sträng till en fil */
rewind(unit);           /* Spola tillbaka en fil till början */
```

Några avslutande synpunkter

Har man nu någon nytta av C på 99:an?
Det beror på.
Om man är intresserad av ett nytt språk för att använda det praktiskt beror det på vilka språk man har använt förut. Själv skulle jag nog hellre använda FORTH på 99:an. På grund av att C-program måste kompileras brukar jag endast använda C om jag har tillgång till en hårddisk och en snabb dator, typ IBM AT eller COMPIS.
För den som inte tycker om FORTH och som skriver i assembler för att få snabba program kan jag rekommendera C. C verkar vara något snabbare än FORTH på 99:an och torde alltså vara det snabbaste språket förutom assembler. Det bör gå betydligt fortare att utveckla program i C än i assembler, även med hänsyn tagen till att kompileringen tar tid.
Den som är nöjd med FORTH kanske vill prova på ett nytt språk. Man bör då vara medveten om kompilatorns

begränsningar och vara beredd på att de C-program man ser i böcker ofta måste skrivas om för att fungera.

För att läsaren av denna artikel skall få en aning hur ett C-program kan se ut visas ett program som skriver ut alla superprimtal. Ett superprimtal är ett primtal som förblir ett primtal även om en siffra i taget plockas bort bakifrån i talet. Exempel: 593 är ett superprimtal eftersom 593, 59 och 3 alla är primtal. Programmet utgår från de ensiffriga primtalen 2, 3, 5 och 7 och lägger på siffrorna 1, 3, 7 och 9 för att skapa tvåsiffriga tal (21, 23, 27, 29, 31, 33, 37, 39 etc). De som är primtal är också superprimtal, och de används för att skapa tresiffriga tal genom att lägga på siffrorna 1, 3, 7 och 9. Primtalen plockas ut och sedan skapas fyrsiffriga tal, etc.

Litteratur

Kernighan, Brian W., och Dennis M. Ritchie: The C Programming Language. Englewood Cliffs, New Jersey. Prentice-Hall, Inc., 1978.

```
#define MAXSUPER 400
main()
{
        static long old[MAXSUPER], new[MAXSUPER] = { 2, 3, 5, 7 };
        int n = 4;
        register int i;

        while (n) {
                print(new, n);
                for (i = 0; i < n; i++)
                        old[i] = new[i];
                n = next(old, new, n);
        }
}

next(old, new, n)
long old[], new[];
int n;
{
        static int digits[] = { 1, 3, 7, 9 } ;
        register int i, j, newn = 0;
        long test;

                for (i = 0; i < n; i++)
                        for (j = 0; j < sizeof digits / sizeof(int); j++) {
                                test = old [i] * 10 + digits[j];
                                if prime(test))
                                        new[newn++] = test;

                        }
        return newn;
}

prime(n)
long n;
{
        long t;

        for (t = 3L; t * t < n && n % t; t += 2)
                ;
        if (n % t != 0L)
                return 1;
        else
                return 0;
}

print(array, n)
long array[];
int n;
{
        register int i;

        for (i = 0; i < n; i++)
                printf("%20ld", array [i]);
}
```

Maximem

Av Lennart Thelander

Maximem är en modul jag köpte från Kanada för en månad sedan. Den modulen innehåller 16KB RAM-minne och 32KB GRAM (Ja just det GRAM). Den innehåller också 8KB ROM. Detta ROM gör att modulen kan fungera som Texas EDITOR/ASSEMBLER modul, men det är inte den stora fördelen. Den stora fördelen är att vilken texasmodul som helst kan läsas av och läggas på en diskett. Sedan kan denna moduls program läggas in i maximem. Maximem uppför sig sedan precis som originalmodulen. Detta gör att man bara behöver en enda modul: Maximem. Jag har nu lagt alla mina moduler på diskett och behöver bara sätta i en diskett och köra. En diskett tar ju mindre plats än en modul och dessutom är ju alla disketter samlade i diskettboxen. Har jag väl laddat in en modul i Maximem finns den kvar ända till jag stänger av datorn (inga batterier alltså). Jag kan också gå in i Maximems motsvarighet till EDITOR/ASSEMBLER och sedan tillbaka till min andra modul utan problem. Maximem har ju helt eget minne som texasmodulen ligger i.

Maximem är gjord i målad plexiglas och sticker ut nästan 15 cm från modulporten. Det är alltså en stor modul. Modulen kostar 209 Kanadensiska dollar, det är ungefär 1200 kronor. Adressen står i PB 85/4 sid 3 längst ner. Med modulen får man också ett "ONE PASS DISK COPY". Detta kan göras i och med att modulen har 48KB RAM (i olika form) och datorn också har 48KB RAM. Det är mer än de 90KB som rymms på en diskett. Jag har inte testat om det går att kopiera dubbelsidiga disketter, men enkelsidiga går med rasande fart jämfört med disk manager. Texasmoduler som är lagda på disk kan kopieras som vanliga filer. Det enda som är litet besvärligt är att flytta bara några moduler till en annan diskett. Då måste en datafil på den nya disketten ändras. Att göra backup av en hel diskett går lika bra som vilken annan diskett som helst.

Vad behöver man nu för utrustning för att använda Maximem? Minnesexpansion och disksystem behövs. Det går visserligen att lagra moduler på kassett utan minnesexpansion, men det kan jag inte rekommendera. Det skulle ta alldeles för lång tid. X-Basic skulle ta teoretiskt 8 minuter och 11 sekunder om kassettspelaren inte stannar någon gång. Det kommer den att göra eftersom X-Basic blir 6 olika filer. För att kunna läsa av en texasmodul behöver man montera in två stycken tryck-knappar i datorn. Den ena är en RESET knapp. Med den kommer man till titelbilden när man än trycker (QUIT går ju att stänga av, det gör inte den här, den fungerar ALLTID). Den andra knappen är en så kallad Load Interrupt Switch. Med den knappen startas moduldump-programmet när texasmodulen har satts in i datorn. Man kan naturligtvis använda denna switch till annat om man kan skriva assemblerprogram. Man kan tex skriva ett skärmdumpprogram som sedan kan startas när som helst genom att trycka på denna knapp. Debuggern till assembler kan också startas på detta sätt när som helst. Det medföljer instruktioner för hur dessa knappar ska kopplas.

Vad som däremot inte följde med som utlovat var GPL-assembler och GPL-disassembler. Jag har skrivit och påpekat detta, men ännu inte fått något svar. På det hela taget är detta en bra modul som jag kan rekommendera alla.

Red: Maximem finns nu även med batteribackup. Det kostar 20 Canada-dollar extra.

Annonser

Säljes:
Mini Memory m. manual 650:–
Fundamentals of TI-99/4A Assembly language.
Ev som dellikvid: minnesexp eller diskkontrollkort
Mikael Jonsson, 046/119700 efter 17.00

Köpes:
P-kodkort
Sören Bernle 042/933 05

Sverige runt

av Niels och Jens Hovmöller

Med Extended BASIC och en joystick kan du kolla om du vet var över 100 svenska "städer" ligger. Med din joystick flyttar du en ring på en Sverige-karta och får sedan besked om hur pass nära du har kommit. Av de 100 poäng som du har från början blir du av med 1, 3, 7 eller 15 om du inte lägger din stad mycket nära dess riktiga läge. När poängen är slut får du börja om från början. Det gäller alltså att hinna besöka så många orter som möjligt innan kapitalet tar slut. 50 städer är ett utmärkt resultat.

Orterna efterfrågas i ungefärlig storleksordning. Förorter till Stockholm och Göteborg har försetts med ett REM, men de kan lätt inkluderas om du vill ha med tex Lidingö eller Mölndal i frågebatteriet.

Det är rätt lätt att lägga in nya städer eller andra punkter (Kebnekaise tex) i programmet. Du kan också göra *Sverige runt* lättare genom att

1) ge dig själv mer än 100 poäng från början (rad 3080) eller

2) minska poängavdraget eller öka de tillåtna felmarginalerna i rad 3470 tom 3500.

Nya städer lägger du in i DATA-satser i början av programmet. Formatet är STAD$,NS,EW där STAD$ är stadens namn med versaler, och NS och EW är läget för en sprite. Tar du bort REM från rad 3240 så ser du vilka värden NS och EW skall ha i de olika DATA-satserna när du kör programet och placerar ringen där staden skall ligga.

Exempel: DATA STORSTAD,150,50

För att skapa kartan har bokstäverna X, Y och Z omdefinierats. Skriv alltså in nya orter i datasatserna med

X som : Y som ; Å som < Ä som = och Ö som >.

Växjö skrivs alltså: "V=:j>".

Med CALL KEY i ställer för CALL JOYST kan programmet ändras så att det inte behövs någon joystick. Då måste också CALL TOUCH-rutinen (rad 3380–3510) ändras.

Städerna på kartan dyker upp och försvinner på ett lite förbryllande sätt, vilket beror på att det inte kan visas mer än fyra sprites på en rad, och både ringen och de tre stora sjöarna är sprites.

100–140 REM-satser
150–190 CALL-satser för programmet
200–300 vinjett
310–1390 DATA-satser med städer och deras lägen
1400–2370 grafik
1500–2180 hexadecimalkoder för kartbilder
2380–2700 kartan
2710–3040 instruktioner
3050–3300 en spelomgång
3310–3340 kartan blir synlig
3350–3370 tom slinga
3380–3510 kolla en gissning
3520–3540 rensar skärmens högra del
3550–3600 slutresultat – ett spel till?
3610–3640 alla städer besökta


```
100 REM SVERIGE RUNT
110 REM av NIELS och JENS                H O V M O L L
    E R
120 REM XBASIC & EN JOYSTICK
130 REM HITTA SVENSKA                    KOMMUNER PA KA
    RTAN
140 REM
150 CALL GRAFIK
160 CALL REGLER
170 CALL SPELA(SN,P,F)
180 CALL RESULTAT(F,Q$)
190 GOTO 170
200 !  **********************
210 !  *                    *
220 !  *    SVERIGE RUNT    *
230 !  *                    *
240 !  * av JENS och NIELS  *
250 !  *                    *
260 !  * H O V M O L L E R  *
270 !  *                    *
280 !  *  A P R I L   8 6   *
290 !  *                    *
300 !  **********************
310 REM DE STORSTA ORTERNA
320 DATA STOCKHOLM,127,67
330 DATA G>TEBORG,152,21
340 DATA MALM>,180,23
350 DATA UPPSALA,119,67
360 DATA V=STER<S,124,58
370 DATA >REBRO,130,48
380 DATA NORRK>PING,140,54
390 DATA J>NK>PING,149,37
400 DATA LINK>PING,141,50
410 DATA H=LSINGBORG,174,22
420 DATA ESKILSTUNA,130,59
430 DATA G=VLE,107,62
440 DATA S>DERT=LJE,131,63
450 DATA KARLSTAD,127,36
460 DATA BOR<S,150,30
470 DATA SUNDSVALL,84,65
480 DATA SKELLEFTE<,51,87
490 DATA V=:J>,161,43
500 DATA LULE<,37,93
510 DATA >RNSK>LDSVIK,75,73
520 DATA LUND,178,26
530 DATA UME<,67,81
540 REM DATA HUDDINGE,128,66
550 DATA KRISTIANSTAD,176,36
560 REM DATA SOLNA,126,67
570 DATA VISB;,156,79
580 REM VISBY AR EN DEL AV               GOTLANDS KOMMUN
590 DATA KALMAR,164,55
600 REM DATA J=RF=LLA,126,66
610 REM DATA NACKA,128,68
620 DATA >STERSUND,68,50
630 REM DATA HANINGE,129,67
640 DATA TROLLH=TTAN,141,31
650 DATA N;K>PING,138,61
660 DATA UDDEVALLA,142,25
670 DATA FALUN,108,48
680 DATA HALMSTAD,165,23
690 REM DATA M<LNDAL,154,210
700 DATA BORL=NGE,110,45
710 DATA SK>VDE,140,40
720 DATA SANDVIKEN,109,58
730 REM DATA BOTK;RKA,128,66
740 DATA V=STERVIK,151,58
750 DATA VARBERG,160,22
760 REM DATA SOLLENTUNA,127,67
770 REM DATA T=B;,126,67
780 DATA NORRT=LJE,121,71
790 DATA KARLSKOGA,127,43
800 DATA MOTALA,138,48
810 DATA KARLSKRONA,176,47
820 DATA HUDIKSVALL,93,63
830 DATA TRELLEBORG,184,27
840 REM DATA LIDING>,127,67
850 DATA LIDK>PING,142,35
860 DATA LANDSKRONA,176,23
870 DATA LUDVIKA,114,45
880 DATA PITE<,42,88
890 DATA N=SSJ>,153,41
900 DATA KATRINEHOLM,133,57
910 REM DET VAR DE FOLK-                 RIKASTE KOMMUN
    ERNA
920 REM NU KOMMER KOMUNER                MED >20000 INV
    ANARE
930 DATA ENK>PING,123,63
940 DATA KARLSHAMN,176,43
950 DATA FALKENBERG,162,21
960 DATA S>DERHAMN,99,62
970 DATA KIRUNA,11,79
980 DATA V=RNAMO,158,37
990 DATA RONNEB;,175,46
1000 DATA MARK KINNA,154,27
```

```
1010 REM DATA SUNDB;BERG,127,67
1020 DATA VETLANDA,157,46
1030 DATA K>PING,124,53
1040 REM DATA DANDER;D,127,67
1050 DATA SIGTUNA,123,67
1060 DATA OSKARSHAMN,159,56
1070 DATA KUNG=LV,151,21
1080 DATA ARVIKA,122,30
1090 DATA KRISTINEHAMN,127,41
1100 DATA AVESTA,114,53
1110 DATA H=RN>SAND,78,68
1120 DATA BODEN,33,91
1130 DATA LJUNGB;,162,33
1140 DATA =NGELHOLM,172,23
1150 REM DATA PARTILLE,153,22
1160 REM DATA LERUM,152,23
1170 DATA ESL>V,177,29
1180 DATA FINSP<NG,136,52
1190 DATA MJ>LB;,141,47
1200 DATA G=LLIVARE,18,83
1210 DATA MARIESTAD,137,40
1220 DATA LINDESBERG,126,48
1230 REM DATA UPPLANDS V=SB;,124,67
1240 DATA ;STAD,184,35
1250 DATA LJUSDAL,92,58
1260 DATA N;BRO,163,51
1270 DATA KUNGSBACKA,154,21
1280 DATA V=NERSBORG,141,31
1290 DATA SOLLEFTE<,73,62
1300 DATA H=SSLEHOLM,173,31
1310 DATA ALINGS<S,147,31
1320 DATA S=FFLE,131,32
1330 DATA FAGERSTA,116,54
1340 DATA TRAN<S,137,49
1350 DATA KRAMFORS,75,65
1360 DATA HUSKVARNA,149,37
1370 DATA FALK>PING,143,35
1380 DATA SALA,119,58
1390 DATA X,1,1
1400 SUB GRAFIK
1410 REM FLAGGAN
1420 CALL CLEAR :: CALL SCREEN(15)
1430 CALL COLOR(1,5,5,2,12,12)
1440 FOR C1=5 TO 7 :: CALL COLOR(C1,2,12):: NEXT C1
1450 CALL HCHAR(1,1,32,24*32):: CALL HCHAR(10,1,40,
     160):: CALL VCHAR(1,10,40,120)
1460 DISPLAY AT(12,5)SIZE(12):"KARTAN(RITAS..."
1470 FOR I=88 TO 89 :: CALL CHARPAT(I,TK$):: CALL C
     HAR(I-30,TK$):: NEXT I
1480 CALL CHAR(60,"0038283847C4444002800384 47C4444
     0028007C4444447C")! SV BOKST
1490 REM
1500 REM TECKENDEFINITION
1510 DATA 03070E1C3870E0C0
1520 DATA C0E070381C0E0703
1530 DATA 1818181818181818
1540 DATA 000000FFFF000000
1550 DATA FFFF000000000000
1560 DATA 0303030303030303
1570 DATA 000000000000FFFF
1580 DATA C0C0C0C0C0C0C0C0
1590 DATA 00000003070E1C18
1600 DATA 000000C0E0703818
1610 DATA 181C0E0703000000
1620 DATA 183870E0C0000000
1630 DATA 030306060C0C1818
1640 DATA 18180C0C06060303
1650 DATA C0C0606030301818
1660 DATA 1818303060 60C0C0
1670 DATA C0F03C0F03000000
1680 DATA 030F3CF0C0000000
1690 DATA 000000C0F03C0F03
1700 DATA 000000030F3FF0C0
1710 DATA E070381F1F3870E0
1720 DATA 18181818 3C7447C3
1730 DATA FFFFC0C0C0C0C0C0
1740 DATA 030303030303FFFE
1750 DATA C0C0C0C0C0C0FFFF
1760 DATA 1818181818 18FFFF
1770 DATA 0000000303030303
1780 DATA 000000000000F8F8
1790 DATA 000000 80C0C0C0C0
1800 DATA F8F0000000000000
1810 DATA 1F1F000000000000
1820 DATA 0000001F1F181818
1830 DATA 1818181F1F000000
1840 DATA 181818F8F0000000
1850 DATA C3C3C3FFFF030303
1860 DATA 03030606EC7E3B1B
1870 DATA 181C0E070F1C3860
1880 DATA 00000087FF783030
1890 DATA 3030303030303F3F
1900 DATA 00000003070EFCF8
1910 DATA 0303E3F31B1F0F07
1920 DATA 303030F0F0381F0F
1930 DATA 1818181818181F1F
1940 DATA 181C1E1B19181818
1950 DATA 03070EDCF8703818
1960 DATA 030306060C0CF8F8
1970 DATA 0000000000000001
1980 DATA 0100000000000000
1990 DATA 0000000000000080
2000 DATA 8000000000000000
2010 DATA 1F1F0C0C06060303
2020 DATA C0E070381C0E0703
2030 DATA C0C0606030301F1F
2040 DATA C0C0C0C000000000
2050 DATA C0C0C0C080000000
2060 DATA 6030180D070E1C18
2070 DATA 0007070303077F7F
2080 DATA 7FFFFFFEC8000000
2090 DATA 80C0E0E0E0F0E0C0
2100 DATA 8080000000000000
2110 DATA 0F7FFF0700000000
2120 DATA 0000000000000000
2130 DATA 00C0F0FFFF000000
2140 DATA 0000000000000000
2150 DATA 0000010307070E1E
2160 DATA 3C3C38F8F0E0C080
2170 DATA 40C080C0C0000000
2180 DATA 0000000000000000
2190 RESTORE 1510
2200 FOR I=33 TO 47
2210 READ TKN$
2220 CALL CHAR(I,TKN$)
2230 NEXT I
2240 FOR I=96 TO 126
2250 READ TKN$
2260 CALL CHAR(I,TKN$)
2270 NEXT I
2280 FOR I=128 TO 135
2290 READ TKN$
2300 CALL CHAR(I,TKN$)
2310 NEXT I
2320 READ A$
2330 CALL CHAR(34,A$)
2340 READ A$
2350 CALL CHAR(127,A$)
2360 CALL CHAR(64,"")
2370 CALL CHAR(140,"00030C1023244848484824231 00C030
     000C03008C4241212121224C40830C000")
2380 CALL MAGNIFY(3)
2390 ! KARTAN I PRINTSATSER
2400 FOR I=1 TO 14 :: CALL COLOR(I,1,1):: NEXT I
2410 PRINT :: PRINT "          &ac     "
2420 PRINT "       kb% )(    "
2430 PRINT "        -    é/    "
2440 PRINT "      Aé    /#    "
2450 PRINT "      hA  d,#     "
2460 PRINT "       (    (p,    "
2470 PRINT "       (  dsn     "
2480 PRINT "     -E   (&      "
2490 PRINT "    ),&E'(-       "
2500 PRINT "    é A( ";CHR$(127);",       "
2510 PRINT "    echEd,        "
2520 PRINT "   -$Ai'(         "
2530 PRINT "   .+tA&          "
2540 PRINT "   & AEü          "
2550 PRINT "   -/ .u$         "
2560 PRINT " Aé+vb/+m         "
2570 PRINT " &l w'jx";CHR$(34)
2580 PRINT "&y+ zfAé          "
2590 PRINT "A//  AgA          "
2600 PRINT " ää!E!( Adm       "
2610 PRINT " öáAA&)m--        "
2620 PRINT " ..''ü#(#é        "
2630 PRINT " E%&'éqEo         "
2640 CALL HCHAR(24,4,38):: CALL HCHAR(24,5,39):: CA
     LL HCHAR(24,6,104)
2650 FOR I=1 TO 4 :: READ T$(I):: CALL CHAR(87+I,T$
     (I)):: NEXT I :: CALL SPRITE(#1,88,1,135,38)::
      ! VANERN
2660 CALL CHAR(88,"1C1D1F0F0F0FC7C7E7FFFFFF76640000
     C0E0F0F0E0F0F0E0C080800000000000")
2670 FOR I=1 TO 4 :: READ T$(I):: CALL CHAR(91+I,T$
     (I)):: NEXT I :: CALL SPRITE(#2,92,1,132,60)::
      ! MALAREN
2680 FOR I=1 TO 4 :: READ T$(I):: CALL CHAR(135+I,T
     $(I)):: NEXT I :: CALL SPRITE(#3,136,1,142,45)
     :: ! VATTERN
2690 CALL NORMALCOLOR
2700 SUBEND
2710 SUB REGLER
2720 DISPLAY AT(8,15):"SL=PP UPP" :: DISPLAY AT(9,1
     5):"ALPHA LOCK"
2730 DISPLAY AT(11,15):"KAN DU "
2740 DISPLAY AT(12,15):"REGLERNA?" :: DISPLAY AT(13
     ,15):"  J N  J" :: ACCEPT AT(13,22)SIZE(-1)BEE
     P VALIDATE("jnJN"):JN$
2750 CALL SUDDA
```

```
2760 IF JN$="j" OR JN$="J" THEN SUBEXIT
2770 DISPLAY AT(3,14):"DU SKA F>RS>KA"
2780 DISPLAY AT(4,14):"PLACERA UT"
2790 DISPLAY AT(5,14):"SVENSKA ST=DER"
2800 DISPLAY AT(6,14):"P< KARTAN"
2810 DISPLAY AT(7,14):"H=RINTILL  "
2820 DISPLAY AT(11,14):"MED ST;RSPAKEN"
2830 DISPLAY AT(12,14):"ST;R DU RINGEN"
2840 DISPLAY AT(13,14):"TILL STADENS"
2850 DISPLAY AT(14,14):"L=GE P< KARTAN"
2860 DISPLAY AT(17,14):"H<LL DEN R>DA "
2870 DISPLAY AT(18,14):"KNAPPEN "
2880 DISPLAY AT(19,14):"INTR;CKT S< G<R"
2890 DISPLAY AT(20,14):"DET SNABBARE"
2900 DISPLAY AT(23,14):"TR;CK ENTER"
2910 ACCEPT AT(23,27)BEEP:Q$ :: CALL SUDDA
2920 DISPLAY AT(3,14):"JU N=RMARE"
2930 DISPLAY AT(4,14):"DU KOMMER"
2940 DISPLAY AT(5,14):"DESTO FLER"
2950 DISPLAY AT(6,14):"PO=NG F<R DU"
2960 DISPLAY AT(7,14):"BEH<LLA"
2970 DISPLAY AT(12,14):"N=R DU TROR"
2980 DISPLAY AT(13,14):"ATT DU LIGGER"
2990 DISPLAY AT(14,14):"R=TT P< KARTAN"
3000 DISPLAY AT(15,14):"TR;CKER DU NED"
3010 DISPLAY AT(16,14):"EN TANGENT"
3020 DISPLAY AT(23,14):"TR;CK ENTER" :: ACCEPT AT(2
     3,27)BEEP:Q$
3030 CALL SUDDA
3040 SUBEND
3050 SUB SPELA(SN,P,F)
3060 CALL SUDDA :: F=0 :: FOR I=5 TO 28 :: CALL COL
     OR(#I,1):: NEXT I
3070 XX=100 :: YY=XX :: CALL SPRITE(#4,140,2,XX,YY)
3080 RESTORE :: P=100
3090 REM ANGE RAD-NR EFTER  RESTORE OVAN OM DU EJ V
     ILL  BORJA MED STOCKHOLM
3100 REM JU HOGRE VARIABELN P SATTS DESTO FLER ORTE
     R    INNAN SPELET TAR SLUT
3110 REM OM P=1545 VISAS                ALLA 103 STADE
     RNA
3120 REM NY STAD
3130 IF P<1 THEN SUBEXIT
3140 F=F+1 :: SN=SN+1 :: READ STAD$,NS,EW :: IF STA
     D$="X" THEN CALL BRAVO
3150 CALL SPRITE(#28-SN,128,1,NS,EW):: DISPLAY AT(8
     ,15):"VAR LIGGER " :: DISPLAY AT(10,15):STAD$;
     "?"
3160 CALL JOYST(1,Y,X)
3170 CALL KEY(1,V,W)
3180 IF V=18 THEN X=X*8 :: Y=Y*8 :: ! FIRE BUTTON
3190 XX=XX-X/4 :: YY=YY+Y/4 :: IF YY>112 THEN YY=11
     2
3200 IF XX<1 THEN XX=1
3210 IF XX>192 THEN XX=192
3220 IF YY<1 THEN YY=1
3230 CALL SPRITE(#4,140,5,XX,YY):: ! RINGEN FLYTTAS
3240 REM DISPLAY AT(13,14):XX;YY :: REM VISAR RINGE
     NS       LAGE PA KARTAN OM DU TAR     BORT REM
     I BORJAN...
3250 REM ...VILKET GOR ATT       DU LATT PLACERAR NY
     A        ORTER PA KARTAN
3260 CALL KEY(0,K,Q)
3270 IF Q=1 THEN CALL TOUCH(SN,P,F):: DISPLAY AT(20
     ,15):"DU HAR";P :: DISPLAY AT(22,15):"PO=NG KV
     AR" :: CALL WAIT :: CALL WAIT :: GOTO 3120
3280 IF Q=1 THEN CALL SOUND(100,880,1):: CALL WAIT
3290 IF P>0 THEN GOTO 3160
3300 SUBEND
3310 SUB NORMALCOLOR
3320 CALL COLOR(1,12,16,2,12,16):: FOR I=3 TO 7 ::
     CALL COLOR(1,2,16):: NEXT I :: FOR I=8 TO 14 :
     : CALL COLOR(I,12,16):: NEXT I
3330 FOR I=1 TO 28 :: CALL COLOR(#I,5):: NEXT I
3340 SUBEND
3350 SUB WAIT
3360 FOR W=1 TO 200 :: NEXT W
3370 SUBEND
3380 SUB TOUCH(SN,P,F)
3390 REM KOLLA HUR BRA       STADEN PLACERATS UT
3400 CALL DISTANCE(#4,#28-SN,I)
3410 FOR TON=1 TO 5 :: CALL SOUND(1,110*2^TON,0)::
     CALL COLOR(#28-SN,7):: CALL COLOR(#28-SN,2)::
     NEXT TON
3420 IF P<1 THEN CALL RESULTAT(F,Q$):: IF P<1 THEN
     IF Q$<>"N" OR Q$<>"n" THEN SUBEXIT
3430 DISPLAY AT(3,15):"DU HAR BES>KT"
3440 DISPLAY AT(4,15):F;"ST=DER"
3450 IF SN=20 THEN SN=1 :: REM NY SPRITE-SERIE
3460 IF I<4 THEN DISPLAY AT(18,15):"BRA" :: SUBEXIT
3470 IF I<10 THEN P=P-1 :: DISPLAY AT(18,15):"SKAPL
     IGT" :: SUBEXIT
3480 IF I<30 THEN P=P-3 :: DISPLAY AT(18,15):"I N=R
     HETEN" :: SUBEXIT
3490 IF I<100 THEN P=P-7 :: DISPLAY AT(18,15):"INTE
     S< BRA" :: SUBEXIT
3500 DISPLAY AT(18,15):"KOLLA KARTAN" :: P=P-15
3510 SUBEND
3520 SUB SUDDA
3530 FOR I=1 TO 24 :: CALL HCHAR(I,15,32,16):: NEXT
     I
3540 SUBEND
3550 SUB RESULTAT(F,Q$)
3560 CALL SUDDA
3570 DISPLAY AT(4,15):"DU HANN" :: DISPLAY AT(5,15)
     :"BES>KA" :: DISPLAY AT(6,15):F;"ST=DER"
3580 DISPLAY AT(8,15):"ETT SPEL TILL?" :: DISPLAY A
     T(9,15):"J N        J" :: ACCEPT AT(9,26)VALID
     ATE("JNjn")BEEP SIZE(-1):Q$
3590 IF Q$="J" OR Q$="j" THEN SUBEXIT ELSE CALL CLE
     AR :: END
3600 SUBEND
3610 SUB BRAVO
3620 CALL SUDDA
3630 DISPLAY AT(8,15):"BRAVO" :: END
3640 SUBEND
```

Redaktören: Som det står i artikeln på sid. 20 så består sjöarna på kartan av sprites. Det visar sig också på den screendump som illustrerar artikeln. ”Hålen” i kartan är sprites.

På rad 3410 i programlistningen finns ett stort tyskt ÿ. Det ska vara ”upphöjt till” (skift 6 på tangentbordet).

Glöm inte medlemsträffen i Stockholm den 13 september!

Här nedan fortsätter listningen av programmet *SORT-DEMO* från sidan 22!

```
4480 PRINT "Du frågas efter nytt antal":"efter avsl
     utad sortering.": :
4490 PRINT "Tillåtet antal är";EMIN;"-";EMAX: : :
4500 GOSUB 3240 :: RETURN
4510 !-----------------------
4520 ! --!--
4530 !
4540 GOSUB 3490
4550 PRINT "Utropstecknet avslutar":"programmet.":
     : : : : :
4560 GOSUB 3240 :: RETURN
4570 !-----------------------
4580 ! --Skifta element
4590 !
4600 SWP(WH)=SWP(WH)+1
4610 X=S(I):: S(I)=S(J):: S(J)=X
4620 RETURN
4630 !é?+
4640 !-----------------------
4650 ! --Färgsättning--
4660 !
4670 SUB FARG(F,B,S)
4680 FOR SATS=0 TO 12 :: CALL COLOR(SATS,F,B):: NEX
     T SATS
4690 CALL SCREEN(S)
4700 SUBEND
4710 !-----------------------
4720 ! --Fördröjning--
4730 !
4740 SUB TID(T,KEY)
4750 FOR I=1 TO T :: CALL KEY(3,KEY,S)
4760 IF S<>0 THEN 4780
4770 NEXT I
4780 SUBEND
4790 !-----------------------
4800 ! --Rutin för jämförelse av ASCII-tecken 32-93
     --
4810 !
4820 SUB COMPARE(S(),I,J,SWAP)
4830 IF S(I)<>93 AND S(J)<>93 THEN 4870
4840 IF S(I)=93 AND S(J)<91 THEN SWAP=1 :: SUBEXIT
4850 IF S(I)=93 THEN SWAP=0 :: SUBEXIT
4860 IF S(I)<91 THEN SWAP=0 :: SUBEXIT ELSE SWAP=1
     :: SUBEXIT
4870 IF S(I)<=S(J)THEN SWAP=0 ELSE SWAP=1
4880 SUBEND
4890 !-----------------------
4900 END
```

Sorteringsdemo

av Bo-Arne Östborg

Programmet demonstrerar och jämför tre olika sorteringsmetoder – Quicksort, Fastsort (= Shellsort) och Bubblesort – vid sortering av slumpgenererade bokstavsgrupper.

Vid sidan av demonstrationen är programmet även nyttigt för inlärning av sorteringsmetoderna genom studium av programlistningen.

Det ursprungliga programmet var skrivet i Microsoft MBasic och fanns i Microcomputing Nov 1983 och omfattas naturligtvis av copyright. Men min version för TI Extended Basic är kraftigt omarbetad, i synnerhet vad gäller skärmpresentationen.

Det fullständiga programmet med REM-satser blir för långt för kassett. Jag har därför gjort versioner både för diskett och kassett. Versionen för kassett har inga REM-satser. Kassettversionen har kvar samma radnummer som diskettversionen. Programlistningen gäller därför för båda versionerna. (Bägge programmversionerna finns i Programbanken.) Den version som är publicerad här är den långa versionen (för diskett).

```
100 ! *****************
110 ! *** SORTDEMO ***
120 ! *****************
130 ! Efter ett program av          Dick Lutz  i
140 ! MICROCOMPUTING 7(1983):11
150 ! Modifierad för
160 ! TI Extended BASIC
170 ! av Bo-Arne östborg
180 ! Version 2. 1986-01-08
190 !************************
200 ! Pre-Scan
210 !
220 GOTO 310 :: CALL CHAR :: CALL COLOR :: CALL MA
    GNIFY :: CALL CLEAR :: CALL SPRITE :: CALL DEL
    SPRITE :: CALL HCHAR :: CALL KEY
230 CALL VCHAR :: CALL SCREEN :: CALL FARG :: CALL
     COMPARE :: CALL TID
240 X,KEY,TAL,SLOW,RI,SS,LNG$,K,T$,LJ,LI,W,FLAG,TO
    P,TOPDISORDER,JJ,J2,J4,N,SWAP,SW,J,J1,I1,F,INI
    T2,TID,WH,RAD,KOD,A$,SPR,NYLEN,INST,I,DINIT
250 T1$,T2$,F$,SP$,SPEED,STATS,BUBB,FAST,QUIK,DSP,
    SLEN,TLEN,EMIN,EMAX,BRAD,FRAD,QRAD,SRAD
260 OPTION BASE 1 :: DIM ST(12,2),S(25),B(25),SWP(
    3),CMP(3)
270 DATA S,O,R,T,D,E,M,O,é
280 !-----------------------
290 !éP-
300 !
310 GOSUB 370 ! Initiering
320 GOSUB 2730 ! Instruktioner
330 RANDOMIZE :: GOTO 480 ! Till huvudprogram
340 !-----------------------
350 ! --Konstanter och startvärden
360 !
370 SRAD=6 :: QRAD=SRAD+2 :: FRAD=SRAD+4 :: BRAD=S
    RAD+6 ! Radnummer för visning på skärm
380 EMAX=25 :: EMIN=3 ! Största och minsta antal t
    ecken att sortera
390 SLEN=20 ! Antal tecken att sortera
400 DSP,QUIK,FAST,BUBB,STATS,SPEED=1
410 SP$="9" :: F$="###" :: A$="FFFFFFFFFFFFFFFF" :
    : CALL MAGNIFY(2)
420 CALL CHAR(128,SEG$(A$,1,6)):: CALL CHAR(136,A$
    ):: CALL CHAR(137,SEG$(A$,1,6)):: CALL COLOR(1
    3,6,12):: CALL COLOR(14,10,12)
430 CALL CHAR(91,"440038447C44444444007C444444447C
    100038447C444444",123,"00440038447C44440044007
    C4444447C00100038447C4444")! Svenska tecken
440 RETURN
450 !-----------------------
460 ! ***Huvudprogram***
470 !
480 DINIT=1 ! Aktivera visning
490 FOR I=1 TO 3 :: CMP(I),SWP(I)=0 :: NEXT I ! No
    llställning av räknare
500 IF INST THEN GOSUB 3000 ! Instruktioner
510 IF NYLEN>0 THEN GOSUB 1940 ! Nytt antal tecken
520 FOR I=1 TO SLEN :: B(I)=INT(RND*29)+65 :: NEXT
     I ! Slumpgenerering av nya tecken att sortera
530 CALL FARG(13,12,10):: CALL CLEAR
540 RESTORE 270
550 FOR SPR=2 TO 9 :: READ A$ :: KOD=ASC(A$):: CAL
    L SPRITE(#SPR,KOD,14,1-16*(SPR>5),SPR*24-12)::
     NEXT SPR
560 GOSUB 2130 :: RAD=SRAD :: WH=0 :: GOSUB 1440 !
     Visa osorterad rad
570 IF QUIK>0 THEN GOSUB 680 :: IF INST THEN 600 !
     Quicksort
580 IF FAST>0 THEN GOSUB 960 :: IF INST THEN 600 !
     Fastsort (Shellsort)
590 IF BUBB>0 THEN GOSUB 1180 ! Bubblesort
600 FOR TID=1 TO 100 :: GOSUB 2400 :: NEXT TID ! F
    ördröjning och samtidig tangentbordsavsökning
610 CALL DELSPRITE(ALL)
620 IF BUBB<1 AND FAST<1 AND QUIK<1 THEN GOSUB 184
    0 ! Välj minst en sorteringsmetod
630 IF BUBB<1 AND FAST<1 AND QUIK<1 THEN BUBB,FAST
    ,QUIK=1 ! Annars utförs alla
640 GOTO 480 ! Sortera ny sträng
650 !-----------------------
660 ! --QUICKSORT
670 !
680 DINIT,INIT2=1 :: RAD=QRAD :: DISPLAY AT(RAD,1)
    :"Q"
690 WH=1 :: GOSUB 1440 :: P=0
700 IF DSP>0 THEN GOSUB 2330
710 I1=1 :: J1=SLEN
720 I=I1 :: J=J1 :: SW=-1
730 IF DSP>0 THEN GOSUB 1440 :: GOSUB 1600 :: GOSU
    B 2330
740 CMP(WH)=CMP(WH)+1
750 CALL COMPARE(S(),I,J,SWAP)
760 IF SWAP=1 THEN GOSUB 4600 ELSE 790
770 IF DSP>0 THEN GOSUB 1540 :: GOSUB 2330
780 SW=-SW
790 IF SW>0 THEN I=I+1 ELSE J=J-1
800 IF DSP>0 THEN GOSUB 1600 :: GOSUB 2330 ELSE GO
    SUB 2400
810 IF I<J THEN 740
820 IF I+1>=J1 THEN 840
830 P=P+1 :: ST(P,1)=I+1 :: ST(P,2)=J1
840 J1=I-1
850 IF DSP>0 THEN GOSUB 2330
860 IF I1<J1 THEN 720
870 IF P=0 THEN 910
880 I1=ST(P,1):: J1=ST(P,2):: P=P-1
890 IF DSP>0 THEN GOSUB 2330
900 GOTO 720
910 GOSUB 1380
920 RETURN
930 !-----------------------
940 ! --SHELL SORT (FASTSORT)
950 !
960 DINIT,INIT2=1 :: RAD=FRAD :: DISPLAY AT(RAD,1)
    :"F"
970 WH=2 :: GOSUB 1440
980 N=SLEN :: J4=N
990 J4=INT(J4/2):: IF J4=0 THEN 1130
1000 J2=N-J4 :: JJ=1
1010 I=JJ
1020 IF DSP>0 THEN GOSUB 2330
1030 J=I+J4
1040 IF DSP>0 THEN GOSUB 1600
1050 CMP(WH)=CMP(WH)+1
1060 CALL COMPARE(S(),I,J,SWAP)
1070 GOSUB 2330
1080 IF SWAP=1 THEN GOSUB 4600 ELSE 1110
1090 IF DSP>0 THEN GOSUB 1540 :: GOSUB 2330
1100 I=I-J4 :: IF I>=1 THEN 1020
1110 JJ=JJ+1 :: IF JJ>J2 THEN 990
1120 GOTO 1010
1130 GOSUB 1380
1140 RETURN
1150 !-----------------------
1160 ! --BUBBLESORT
1170 !
1180 DINIT,INIT2=1 :: RAD=BRAD :: TOPDISORDER=SLEN
     :: DISPLAY AT(RAD,1):"B"
1190 WH=3 :: GOSUB 1440
1200 TOP=TOPDISORDER :: FLAG=0 :: I=1 :: J=2
1210 IF DSP>0 THEN GOSUB 1600 :: GOSUB 2330
1220 CMP(WH)=CMP(WH)+1
1230 CALL COMPARE(S(),I,J,SWAP)
1240 IF SWAP=1 THEN GOSUB 4600 ELSE 1270
1250 FLAG=1 :: TOPDISORDER=I
1260 IF DSP>0 THEN GOSUB 1790
1270 GOSUB 2330
1280 IF J>=TOP THEN IF FLAG=0 THEN 1330 ELSE TOP=TO
     PDISORDER :: GOTO 1200
1290 IF DSP>0 THEN GOSUB 2330
1300 I=I+1 :: J=I+1
1310 IF DSP>0 THEN GOSUB 1600 :: GOSUB 2330
1320 GOTO 1220
1330 GOSUB 1380
1340 RETURN
1350 !-----------------------
1360 ! --Sortering klar
1370 !
1380 GOSUB 1440 :: CALL HCHAR(RAD+1,6,32,25)! Visa
     den sorterade raden
```

```
1390 IF STATS>0 THEN GOSUB 1730 ! Visa antal jämför
     elser och omflyttningar
1400 RETURN
1410 !-----------------------
1420 ! --Visa nuvarande ordning
1430 !
1440 IF DINIT<>1 THEN 1460
1450 FOR W=1 TO SLEN :: S(W)=B(W):: NEXT W :: DINIT
     =0
1460 FOR W=1 TO SLEN
1470 CALL HCHAR(RAD,5+W,S(W))
1480 IF WH=1 THEN IF I1<J1 THEN IF W>=I1 AND W<J1 T
     HEN CALL HCHAR(RAD+1,5+W,137)ELSE CALL HCHAR(R
     AD+1,5+W,32)
1490 NEXT W
1500 RETURN
1510 !-----------------------
1520 ! --Visa endast ändring
1530 !
1540 CALL HCHAR(RAD,5+I,S(I)):: CALL HCHAR(RAD+1,5+
     I,128)
1550 CALL HCHAR(RAD,5+J,S(J)):: CALL HCHAR(RAD+1,5+
     J,128)
1560 RETURN
1570 !-----------------------
1580 ! --Visa nuvarande I & J
1590 !
1600 IF INIT2=1 THEN INIT2=0 :: GOTO 1650
1610 IF I<>LI THEN CALL HCHAR(RAD+1,5+LI,32)
1620 IF J<>LJ THEN CALL HCHAR(RAD+1,5+LJ,32)
1630 IF WH>1 THEN IF I<>LI THEN CALL HCHAR(RAD,5+LI
     ,S(LI))
1640 IF WH>1 THEN IF J<>LJ THEN CALL HCHAR(RAD,5+LJ
     ,S(LJ))
1650 IF WH>1 THEN CALL HCHAR(RAD,5+I,S(I))
1660 IF WH>1 THEN CALL HCHAR(RAD,5+J,S(J))
1670 IF I=J THEN CALL HCHAR(RAD+1,5+I,42)ELSE CALL
     HCHAR(RAD+1,5+I,94):: CALL HCHAR(RAD+1,5+J,94)
1680 LI=I :: LJ=J
1690 RETURN
1700 !-----------------------
1710 ! --Visa status efter sortering
1720 !
1730 DISPLAY AT(RAD+9,18):USING F$:CMP(WH)
1740 DISPLAY AT(RAD+9,24):USING F$:SWP(WH)
1750 RETURN
1760 !-----------------------
1770 ! --Visa endast förändring vid bubbelsortering
1780 !
1790 CALL HCHAR(RAD+1,5+I,128):: CALL HCHAR(RAD+1,5
     +J,128)
1800 RETURN
1810 !-----------------------
1820 ! --Val av sorteringsmetod
1830 !
1840 T$="C" :: GOSUB 4000 :: CALL TID(3000,K):: CAL
     L CLEAR :: CALL DELSPRITE(#1):: IF K=-1 THEN 1
     900
1850 T$=CHR$(K)
1860 IF T$="I" THEN GOSUB 3000
1870 IF T$="B" THEN BUBB=1
1880 IF T$="Q" THEN QUIK=1
1890 IF T$="F" THEN FAST=1
1900 RETURN
1910 !-----------------------
1920 ! --Ny längd
1930 !
1940 LNG$="" :: NYLEN=0
1950 CALL CLEAR :: DISPLAY AT(5,1):"ANTAL BOKSTÄVER
      ATT SORTERA:": : :"Tillåtet antal är ";STR$(E
     MIN);"-";STR$(EMAX);"."
1960 DISPLAY AT(10,1):"Nuvarande antal:";SLEN :: DI
     SPLAY AT(14,1)BEEP:"Nytt antal:"
1970 FOR TID=1 TO 500
1980 CALL KEY(3,K,SS)
1990 IF SS=0 THEN 2020
2000 IF K=13 THEN 2070 ELSE IF K=7 THEN 1940
2010 IF K>=48 AND K<=57 THEN 2040
2020 NEXT TID
2030 GOTO 2070
2040 DISPLAY AT(14,14+LEN(LNG$)):CHR$(K):: FOR I=1
     TO 200 :: NEXT I
2050 LNG$=LNG$&CHR$(K)
2060 GOTO 1970
2070 IF LNG$<>"" THEN TLEN=VAL(LNG$)ELSE 2090
2080 IF TLEN<EMIN OR TLEN>EMAX THEN 1940 ELSE SLEN=
     TLEN
2090 RETURN
2100 !-----------------------
2110 ! --Förbered skärmen
2120 !
2130 GOSUB 2260
2140 FOR RI=15 TO 21 :: CALL HCHAR(RI,16,32-104*(ST
     ATS<0),16):: NEXT RI
2150 CALL VCHAR(1,1,136,24):: CALL VCHAR(1,32,136,2
     4)
2160 CALL HCHAR(14,2,136,30):: CALL HCHAR(15,2,136,
     13):: CALL HCHAR(21,2,136,13):: CALL HCHAR(22,
     2,136,30):: CALL VCHAR(15,15,136,7)
2170 IF STATS<0 THEN 2230
2180 DISPLAY AT(15,16):"Jämför Omflyt" :: CALL HCHA
     R(15,31,116)
2190 CALL HCHAR(16,18,45,6):: CALL HCHAR(16,25,45,7
     )
2200 IF QUIK>0 THEN DISPLAY AT(17,15):"Q"
2210 IF FAST>0 THEN DISPLAY AT(19,15):"F"
2220 IF BUBB>0 THEN DISPLAY AT(21,15):"B"
2230 DISPLAY AT(23,1):"Siffertangenterna (0-9) styr
     " :: DISPLAY AT(24,1):"hastigheten, som nu är"
      :: CALL HCHAR(24,27,ASC(SP$))
2240 RETURN
2250 !
2260 IF INST THEN T1$="Anv. visas" :: T2$="efter so
     rt." ELSE T1$="Tryck I för" :: T2$="anvisninga
     r"
2270 DISPLAY AT(17,1)SIZE(11):T1$
2280 DISPLAY AT(19,1)SIZE(11):T2$
2290 RETURN
2300 !-----------------------
2310 ! --Hastighet
2320 !
2330 GOSUB 2400
2340 IF SPEED=1 THEN 2360
2350 FOR SLOW=1 TO SPEED*10 :: GOSUB 2400 :: NEXT S
     LOW
2360 RETURN
2370 !-----------------------
2380 ! --Tangentbordsavsökning
2390 !
2400 CALL KEY(3,K,SS)
2410 IF SS=0 THEN 2560
2420 T$=CHR$(K)
2430 IF T$<="9" AND T$>="0" THEN GOSUB 2600 :: GOTO
      2560
2440 IF T$="D" THEN DSP=-DSP :: CALL HCHAR(RAD+1,5,
     32,25):: GOTO 2560
2450 IF T$="X" THEN STATS=-STATS :: GOSUB 2130 :: G
     OTO 2560
2460 IF T$="B" THEN BUBB=1 :: GOTO 2560
2470 IF T$="Q" THEN QUIK=1 :: GOTO 2560
2480 IF T$="F" THEN FAST=1 :: GOTO 2560
2490 IF T$="A" THEN BUBB,QUIK,FAST=1 :: GOTO 2560
2500 IF T$="C" THEN BUBB,QUIK,FAST=-1 :: GOTO 2560
2510 IF T$="P" THEN DISPLAY AT(1,25):"Paus" :: GOSU
     B 2670 :: DISPLAY AT(1,25):" " :: GOTO 2420
2520 IF T$="L" THEN NYLEN=1 :: DISPLAY AT(21,1)SIZE
     (12):" L-ändring" :: DISPLAY AT(22,1)SIZE(12):
     " efter sort." :: GOTO 2560
2530 IF T$="I" THEN INST=1 :: GOSUB 2260 :: GOTO 25
     60
2540 IF T$=CHR$(6)THEN CALL DELSPRITE(ALL):: CALL C
     LEAR :: GOTO 480
2550 IF T$="!" THEN CALL CLEAR :: GOTO 4900
2560 RETURN
2570 !-----------------------
2580 ! --Hastighetsändring
2590 !
2600 TAL=K-48
2610 SPEED=16-INT(5*SQR(TAL))
2620 SP$=T$ :: CALL HCHAR(24,27,K)
2630 RETURN
2640 !-----------------------
2650 ! --Paus
2660 !
2670 CALL KEY(3,K,SS)
2680 IF SS=0 THEN 2670
2690 RETURN
2700 !-----------------------
2710 ! --Instruktioner
2720 !
2730 CALL FARG(1,1,4):: CALL CLEAR
2740 PRINT TAB(5);"S O R T -- D E M O": : :
2750 PRINT "Detta program visar 3 olika":"sortering
     salgoritmer": :" -Bubble sort":" -Quicksort":"
      -Fastsort": :
2760 PRINT "och jämför dem.": :
2770 PRINT "Programmet startar direkt":"med förvald
     a funktioner, men":"tillåter ändringar av dem"
     :"under drift.": : :
2780 PRINT "Tryck ned en tangent för":"instruktione
     r."
2790 CALL FARG(2,16,4):: CALL TID(2000,KEY)
2800 CALL FARG(1,1,4):: CALL CLEAR
2810 PRINT "Följande är styrtangenter:": :
2820 PRINT "    Q Väljer Quicksort"
2830 PRINT "    F Väljer Fastsort"
2840 PRINT "    B Väljer Bubblesort"
2850 PRINT "    A Väljer Alla (Q,F,B)"
2860 PRINT "    C Upphäver val"
2870 PRINT "    D Visning av sortering"
2880 PRINT "    P Paus"
2890 PRINT "    X Visning av statistik"
```

```
2900 PRINT "   L Ändring av antal tecken"
2910 PRINT " 0-9 Hastigheten"
2920 PRINT "REDO Aterstartar programmet"
2930 PRINT "   ! Bryter programmet": : :
2940 PRINT "Tryck ned en tangent för":"start."
2950 CALL FARG(2,16,4):: CALL TID(2000,KEY):: CALL
     CLEAR
2960 RETURN
2970 !----------------------
2980 ! --Visa huvudinstruktion och därefter underin
     struktion
2990 !
3000 INST=0 :: GOSUB 3300
3010 CALL TID(3000,K)
3020 IF K=-1 THEN 3190
3030 T$=CHR$(K)
3040 IF T$="I" THEN GOSUB 3300 :: GOTO 3010
3050 IF T$>="0" AND T$<="9" THEN GOSUB 3540 :: GOTO
      3010
3060 IF T$="Q" THEN GOSUB 3620 :: GOTO 3010
3070 IF T$="B" THEN GOSUB 3800 :: GOTO 3010
3080 IF T$="F" THEN GOSUB 3870 :: GOTO 3010
3090 IF T$="R" THEN 3190
3100 IF T$="A" THEN GOSUB 4100 :: GOTO 3010
3110 IF T$="C" THEN GOSUB 4000 :: GOTO 3010
3120 IF T$="D" THEN GOSUB 4220 :: GOTO 3010
3130 IF T$="P" THEN GOSUB 4300 :: GOTO 3010
3140 IF T$="X" THEN GOSUB 4370 :: GOTO 3010
3150 IF T$="L" THEN GOSUB 4450 :: GOTO 3010
3160 IF T$=CHR$(6)THEN GOSUB 3940 :: GOTO 3010
3170 IF T$="!" THEN GOSUB 4540 :: GOTO 3010
3180 GOTO 3010
3190 T$="" :: CALL CLEAR :: CALL DELSPRITE(#1)
3200 RETURN
3210 !----------------------
3220 ! --Textsubrutin 1
3230 !
3240 PRINT :"Tryck I för huvudinstruktion"
3250 PRINT "Tryck R för fortsatt drift" :: CALL FAR
     G(2,15,13)
3260 RETURN
3270 !----------------------
3280 ! --Huvudinstruktion
3290 !
3300 T$=" " :: GOSUB 3490
3310 PRINT "Tryck resp. tangent för":"förklaring av
      funktionen:": :
3320 PRINT "   A-Alla sorteringsmetoder"
3330 PRINT " QFB-Val av sortering"
3340 PRINT "   C-Upphäver val"
3350 PRINT "   D-Visning av sortering"
3360 IF DSP>0 THEN CALL HCHAR(23,3,42,2)
3370 PRINT "   P-Paus (Annan tangent":"     bryter)
     "
3380 PRINT "   X-Visning av statistik"
3390 IF STATS>0 THEN CALL HCHAR(23,3,42,2)
3400 PRINT "   L-Ändring av antal tecken"
3410 PRINT " 0-9-Hastighetsändring"
3420 PRINT "REDO-Aterstartar programmet"
3430 PRINT "   !-Avslutar programmet": : :
3440 PRINT "** ovan visar, att":"   funktionen är a
     ktiv": : :
3450 GOSUB 3250 :: RETURN
3460 !----------------------
3470 ! --Subrutin för underinstruktioner
3480 !
3490 CALL FARG(1,1,13):: CALL CLEAR :: CALL SPRITE(
     #1,ASC(T$),5,17,120)
3500 RETURN
3510 !----------------------
3520 ! --0 till 9
3530 !
3540 GOSUB 3490
3550 PRINT "Siffrorna 0 till 9 styr":"sorteringshas
     tigheten.": :"Högre siffra ger högre":"hastigh
     et.": : :
3560 PRINT "Nuvarande hastighet är ";SP$;".": : :
3570 PRINT "För långsammare sortering än":"0 kan pa
     us (P) väljas.": : :
3580 GOSUB 3240 :: RETURN
3590 !----------------------
3600 ! --Q--
3610 !
3620 GOSUB 3490
3630 PRINT TAB(5);"aktiverar Quicksort.": :
3640 IF QUIK>0 THEN GOSUB 3740 ELSE GOSUB 3750
3650 GOSUB 3690 :: GOSUB 3240 :: RETURN
3660 !----------------------
3670 ! --Textsubrutin 2
3680 !
3690 PRINT "För att välja bort metoden,":"måste Du
     upphäva alla val":"med C och därefter välja på
     ":"nytt. Alla kan väljas med A.": :
3700 RETURN
3710 !----------------------
3720 ! --Textsubrutin 3
3730 !
3740 PRINT "Den är nu aktiverad.": : :: GOTO 3760
3750 PRINT "Den är nu icke aktiverad.": :
3760 RETURN
3770 !----------------------
3780 ! --B--
3790 !
3800 GOSUB 3490
3810 PRINT TAB(5);"aktiverar Bubblesort.": :
3820 IF BUBB>0 THEN GOSUB 3740 ELSE GOSUB 3750
3830 GOSUB 3690 :: GOSUB 3240 :: RETURN
3840 !----------------------
3850 ! --F--
3860 !
3870 GOSUB 3490
3880 PRINT TAB(5);"aktiverar Fastsort.": :
3890 IF FAST>0 THEN GOSUB 3740 ELSE GOSUB 3750
3900 GOSUB 3690 :: GOSUB 3240 :: RETURN
3910 !----------------------
3920 ! --REDO--
3930 !
3940 CALL FARG(1,1,13):: CALL DELSPRITE(#1):: CALL
     CLEAR :: PRINT TAB(12);"REDO": : : : : :"avbry
     ter sorteringen och":"återstartar programmet."
3950 PRINT "Om I eller L trycks före":"REDO, nås de
     ssa funktioner":"direkt utan att sorteringen":
     "först avslutas.": : : : :
3960 GOSUB 3240 :: RETURN
3970 !----------------------
3980 ! --C--
3990 !
4000 GOSUB 3490
4010 PRINT "inaktiverar alla sorterings-":"metoder.
      Du måste därefter":"välja på nytt.": :
4020 PRINT "Välj Quicksort   med Q"
4030 PRINT "Välj Fastsort    med F"
4040 PRINT "Välj Bubblesort  med B"
4050 PRINT "Välj Alla tre    med A": :
4060 GOSUB 3240 :: RETURN
4070 !----------------------
4080 ! --A--
4090 !
4100 GOSUB 3490
4110 PRINT "aktiverar alla tre sorte-":"ringsmetode
     rna.": :
4120 PRINT "Följande metoder är nu":"aktiverade:";
4130 IF QUIK>0 THEN PRINT " Q";
4140 IF FAST>0 THEN PRINT " F";
4150 IF BUBB>0 THEN PRINT " B";
4160 IF QUIK<0 AND FAST<0 AND BUBB<0 THEN PRINT " I
     ngen.": :ELSE PRINT : :
4170 PRINT "Om Du vill inaktivera någon,":"måste Du
      bryta alla med C":"och därefter välja på nytt
     .": :
4180 GOSUB 3240 :: RETURN
4190 !----------------------
4200 ! --D--
4210 !
4220 GOSUB 3490
4230 PRINT "kopplar till/från visningen":"av sorter
     ingsförloppet.": :
4240 IF DSP>0 THEN GOSUB 3740 ELSE GOSUB 3750
4250 PRINT "Sorteringen går fortare utan":"visning.
     ": : :
4260 GOSUB 3240 :: RETURN
4270 !----------------------
4280 ! --P--
4290 !
4300 GOSUB 3490
4310 PRINT TAB(8);"ger en paus.": :
4320 PRINT "Både sortering och visning":"stoppas. D
     u kan fortsätta":"genom att trycka godtycklig"
     :"tangent (utom P) inkl.":"styrtecken.": :
4330 GOSUB 3240 :: RETURN
4340 !----------------------
4350 ! --X--
4360 !
4370 GOSUB 3490
4380 PRINT "kopplar till/från statistik-":"rapporte
     n.": :
4390 IF STATS>0 THEN GOSUB 3740 ELSE GOSUB 3750
4400 PRINT "Denna rapport visar hur":"många jämföre
     lser och om-":"flyttningar, som behövdes.": :
4410 GOSUB 3240 :: RETURN
4420 !----------------------
4430 ! --L--
4440 !
4450 GOSUB 3490
4460 PRINT "ger möjlighet att ändra":"antalet eleme
     nt att sortera.": :
4470 PRINT "Nuvarande antal är";SLEN
```

Programlistningen fortsätter på sidan 19 (längst ner i högerspalten)

Programbanken – ny förändrad lista

Här visas en lista på de tillgängliga programmen i PROGRAMBANKEN. Programmen kan köpas! Priset är uppdelat i *två delar*, dels en startkostnad som täcker kostnaden för porto och datamediet, dels en kopieringsavgift för *varje* program.

KASSETT	Startkostnad	35:- per kassett
	Kopieringsavgift	15:-
DISKETT	Startkostnad	55:- per diskett
	Kopieringsavgift	10:-

OBS! Var noggrann när DU anger programnumret. Det är det jag går efter vid kopieringen

Som förut får DU tre program i utbyte när DU skickar in ett program *DU SJÄLV HAR GJORT!*

Ange om ditt program är skrivet i TI-Basic, Extended Basic, Assembler, Pascal eller i Forth. Beskriv noggrant vilken utrustning som behövs (även i själva programmet), hur man kör programmet och vad som behöver göras för att programmet skall fungera. Skriv även vilken modul som erfordras till programmet.

Kodförklaring

De fyra första siffrorna är interna beteckningar (diskettnummer, programkategori).

Femte siffran anger ursprungsland:

1 = England, USA 2 = Holland, Belgien, Tyskland
3 = Sverige 7 = Finland

De tre sista siffrorna är ett internt löpnummer.

Bokstaven, som finns sist, anger vilket språk som används i programmet

E = Engelska, F = Franska, H = Holländska

Om bokstav saknas så är ledtexten på svenska eller så finns det ingen ledtext.

Dessutom kan ett D finnas allra sist i programnumret och det anger att datafil(er) ingår.

Eventuella bokstäver efter programnumret talar om vilken utrustning som krävs.

C = CALL FILES(1) måste användas om man har diskettenhet.

D = Diskettenhet
F = Forth
M = MiniMemory
P = Printer
J = Joystick
T = Terminal Emulator II
E = Minnesexpansion
A = Editor/Assembler
X = Extended-Basic
S = Speech Synthesizer (pratorn)
K = Personal Record Keeping

Programkategorier

02 = Operativsystem
10 = Registerprogram
12 = Ordbehandling
14 = Ekonomi
20 = Regression, Kurvanpassning
21 = Statistik, Varians
29 = Statistik, Sannolikhet
30 = Linjär algebra
39 = Allmän matematik
65 = Elektronik
78 = Astronomi
87 = Assemblerrutiner
88 = Assemblerprogram
89 = Subrutiner, subprogram
90 = Basicprogram
91 = Spel
92 = Utbildning
96 = Musik
97 = Demo
99 = Övrigt

29 = Statistik, Sannolikhet
30 = Linjär algebra
39 = Allmän matematik
65 = Elektronik
78 = Astronomi
87 = Assemblerrutiner
88 = Assemblerprogram
89 = Subrutiner, subprogram
90 = Basicprogram
91 = Spel
92 = Utbildning
96 = Musik
97 = Demo
99 = Övrigt

02 Operativsystem

01101001E --X Släktforskningsregister
01101002E --- Adressregister
11103069 --- Filläsning på Kassett
11103072 --K Snabbfil till Personal Record Keeping modulen. PB 85-2
14103045 PDX Telefonregister. PB 84-4
15103092 -DX Tel-Ad. Telefonadresser
17103012 --- Nummersändaren. PB 83-4/5

12 Ordbehandling

01121001E --P Word-processor
10121010E --C Ordbehandling för TP-printer
11123064 --- P:TEXTIN. PB 84-3
11123065 --- P:TEXTUT, Lista P:TEXTIN filer. PB 84-3
14123039 --P Nova-Verba
16123007E --X LT-Miniwriter. PB 85-1
17123006 --- P:KOPIERA. PB 86-2

14 Ekonomi

07141001H --X Annuities
12143006 --- Ränteberäkningar

20 Regression, kurvanpassning

01201001E --- Uträkning av en linjes ekvation efter ett antal punkter fördelade kring en linje
01201002E --- Uträkning av riktningskoeff

21 Statistik, varians

06211001E --C Beräkning av standardavvikelser

29 Statistik, sannolikhet

09291001H --- Statistiska kombinationer etc

30 Linjär algebra

06301001E --- Matris invertering/multiplikation
14303050 --X Lösning av linjära ekvationssystem med Gauss-eliminationsmetod

39 Allmän matematik

01391001E --- Integral, derivata, andragradsekv reella rötter, cp analys, definiera chars, 3-D plot, fakultet och primtalstest
01391002E --- Lösning av ekvationssystem
01392001E --- Division med valfritt antal decimaler i svaret
09391003H --- Differential eq
11393071 --- Rita sinuskurva. PB 85-2
12393005 --- Matrisberäkningar, lösning av ekvationssystem
14393051 --X Math-Pac, Integration, Newton-Raphson, Intervallhalvering, Romberg-integration

65 Elektronik

01651001E --- Beräkning av komponentvärden för resistiv parallellkoppling, kondensator i serie, resonans för spole och kondensator, omvandling frekvens - våglängd, ohm's lag, uträkning av antennlängd
16653004 --- Ohmslag, räknar ut ampere, ohm, volt och watt. PB 85-3

78 Astronomi

01781001E --- Beräkning av geostationära satelliters positioner
15783000 --X Trippelpunkten på 99:an, Tyngdpunkten till en sfärisk triangel. PB 85-1
16783012 --- Stjärnkikeri. PB 86-2

87 Assemblerrutiner

15873002 EAD K:INPUT, subprogram för inmatning av data från tangentbordet. Källkod. PB 86-2

88 Assemblerprogram

15883001 EAD Disassembler. PB 84-4. Ersatt av diskett
16883011 EAD CSAVE, minnesdumpning till kassett. PB 85-2

89 Subrutiner, subprogram

12893055 --- Screen-Ram. PB 84-5
13891057 EAD Cursor, ändra utseendet på cursorn
13891058 --- Ensa-komma. PB 84-5
14893034 --- Teckendefinierare
15893097 --- Snabbare grafik med TI-Basic. PB 84-1
16893003 -EX Miniassembler. Utmaningen 85-2
16893014 --- Datain-/utmatning kassett. PB 85-3

90 BASIC-program

06901003E --X Sortering och utskrift från flera disketter av upp till 300 program
06901004E --- Diskkatalog med utskrift
07902002E --P Banner, stor text utskriven på TP-printer
08901006E PEX Utskrift av titelsida för ex listningar
08902003E --- Stora tecken för 12x28 teckenskärm
10901008E --- Very large characters
11901011 --- Stapeldiagram
11903063 PDX Lista Basicprogram med Basicprogram. PB 84-4
11903066 -PX PGM LIST59, Lista TI59 program på TI99/4A. PB 84-2
11903067 --- Screen-Saver, spara skärmen. PB 84-3

11903068 --- Testbild för Din Dator. PB 84-3
11903075 -DX LISTA ORD. Listar DIS/VAR 80-filer med 55 tecken/rad. PB 85-1
12901054 -JX Color-Draw, Ritprogram med färger. PB 84-5
13903022 -DX Fil-Data-Editor, för DIS/VAR 80 filer
13903027 --X Grafik Exklusive, Kassettbaserat
13903060 PDX List28, Lista program 28tkn/bredd. PB 85-3
13903062 -DX P:TILLCOMP, överför P:TEXTIN filer till Companion format. PB 84-4
14901053 --X Talkonvertering, Hex-Dec-Binärt. PB 84-5
14902052 --X Färg-Editor. PB 84-5
14903042 --- Kassettinnehåll. PB 84-4
15901089E -DX Merge/Read, listar DIS/VAR 163 filer
15901090E -EX Mg/Peeker, listar minnes-adresser
15903005 --- Baskonvertering, Hex-Dec-Bin. PB 84-1
15903076 PDX Listprogram, Listar Basicprogram
15903077 -SX Trunkering, skapa egna ord med trunkering. PB 84-4
15903088 -EX List-Size, listar storleken på dina program
15903093 -DX Char-Adapt, Teckenändring, lagras som DIS/VAR 80 filer
15903094 -JX Seppo Huikuris Sprite-Maker. PB 84-1
15903095 EDX Load-Assembler-Bas, omformar Assembler-program till Extended Basic program. PB 84-1
15903098 -JX Tommy Erikssons Sprite-Maker. PB 86-1
16901015E -EX Generell programstartare. PB 85-1
16903006 -DM Diskkatalog i Textmode. PB 84-4
16903016 --D CATALOG. Listar diskettkatalogen på skärmen. PB 85-1
16903018 --X Polsk notation. PB 84-2
17903004 --M Mini-ASMBAS. PB 86-2
17903011 -JX Kalkylator med 99:an. PB 86-1

91 Spel

01911028H --- Car driver
01911029H --- Animals
01911030H --- Space Invaders
02911001E --- Master Mind
02911002E --X Prickskytte med kanon mot en kyckling
02911003E --- Startrek, textadventure
02911006E --- Othello
02911007E --- Robotjakt
02911008E --- 15-spel
02911009E --- Tärningsspel
02911011E --- Keno
02911012ED --D Star Guard
03911013E --- Miner
03911014E --C Yahtzee
03911015E --- Backgammon
03911016E --- 3D Tic-tac-toe
03911017E --X Black-Jack II
03911018E -SX Not one
03911019E --X Datorpoker
03911024E --- Fyra i rad (luffarschack)
04911025E --- Startrek
04911026E --X Hjälp kycklingen över vägen
04911027E --- Katapult
04911031E --- ISOLA. Isolera motståndaren
06911032E --- Vem skjuter den sista roboten
06911034H --X L-game
06912004E --- Slalom
06912005E --- Killer
07911035H --X Wari
07911036H --- Ta dig fram i en osynlig labyrint
07911037H --X Eliza
07911040H --X Black jack
07911041H --X Space battle
07911043H --- Enarmad bandit
07911045E --- Labyrint. 2 spelare
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 --- Breakout, bollspel squashtyp
09911059H --J Ritprogram i olika färger
09911060H -JX Treasure hunt, labyrintspel
09912008F --X Asteriod (tal)
09912009E --- Battlestar
10911061H --X Kermit
10911062H -JX Find the gun
10911063E --- Matematikspel
10911065E --- Camel, adventureritt i öknen
11911055 --X Hängning
11917001 -JX Inkräktarna
11917002 -JX Päron
11917003 -JX Masken
12911001 --X One Check, brädspel
12913002 -JX Jockes Hostility
12913004 --X Giljotinen, ordlek
12913007 --- Tjugoett
12913009E --- Octopus
12913012 --- The Hunt of Aladin
12913013 --- Bowling
12913014 --- Bilrally i Norge
12913015 --- Masken II
12913016 --- Djungelfight
12913017 -JX Snake, Ormen
12913018 -JX Zombie
13913019 --J Kami-Kaze
13913020 -JX Ubåt
13913023 --- Jumping Jack
13913024E --- Astrostorm
13913025 --- Goblin
13913026E --- Identify the States
14911037E --X One Check II
14911046E --X Pricka flygplan
14911048E --X Spacegame
14913035 --X Skroting
14913036 -JX Starport 99
14913040E --J Blockman
14913044 --X Mastermind
15913079 --X Adventurespel, Gör dina egna Adventurespel. PB 84-3
16913001 --- Helikopter. PB 86-2
16913002 --X Lander, labyrintspel. PB 85-4
16913003 --X Dart. PB 85-4
16913005 --J Ormar, (liknar Masken). PB 85-4
16913008 -JX NUC, Atombombsspel. Försvara staden PB 85-4
16913013 --X TOTO-G. Galoppspel. PB 85-2
17913001 --- Hangman. PB 83-4/5
17913002 --- Sifferjakt. PB 83-4/5
17913005 --M Masken för Mini Memory. PB 86-2
17913010E --X Korsord. PB 86-1

92 Utbildning

01921006H --X Årvsanlag
04921001E --- Enkla räkneövn. med de fyra räknesätten
04921002E --- Kemifrågor
04921003E --X Relative IQ-test
04921004E --- Algebra
04921005E --- Projectile problems, beräkn.-spel
07921007E --- Träning i dåtid (imperfekt)
10921008E --- Räkneträning med bråkdelar
13923029 --X Huvudräkning
13923030 --X Glosträning, kassettbaserad
13923031 EDX Byggnadsmekanik
16923009 EDX Glosträning, diskbaserad

96 Musik

05962001 --C Johann Strauss. Tales from Vienna woods
05962002 --- Never on a sunday
05962003 --X Berceuse
05962004 --X Beethoven. Symfoni nr 5
05962005 --X Serenade
05962006D -DX Amazing graze
05962007 -CX Beethoven opus 27
05962008D -DX Time in a bottle
05962009D -DX You light up my life
05962010 --- Bummble-boogie
06962001 --- The pink panther
06962006 --- Let me call you sweetheart
06962011D --D Fiddler on the roof
06962012 --- Bach 3. menuet
07962013 --- I'm looking through you
08961001E --X Killing me softly
08962014 --X Chopin op 11 # 3
12963056 --- Tremolo. PB 84-5
14963033 --- TI-Keyboard
14963043E --- When the Saints ...
16963019 --- Ta det piano. PB 84-2
17963003 --- Musicbox. Melodierna Bä bä vita lamm, Vedhuggaren, Blinka lilla stjärna och London bridge
17963008 --X Sätt musik till dina X-Basicprogram. PB 86-2
17963009 --X Inledningen till Gloriasatsen "Iste confessor" av Palestrina. PB 86-2

97 Demo

01971001H --C Demo av färger, grafik och ljud
06971002E --- Kaleidoscope
06972004 --X Halloween (sprites)
06972005 --X Sprites
07971003H --- Demonstration av olika teckenstorlekar
11973070 -EX Demoladdning av Assembler program
13973059 --- Korsord
14973038 --X Mönstergrafik
15971091 -EX Mg/Text, Textmode
15973003 --M Textmode i TI-Basic. PB 84-4
15973004 --M Auto-Sprite i TI-Basic. PB 84-4
15973099 --X Train. Demonstrerar doppler-effekt. PB 85-2
16973010 EAD Tapetest. PB 85-2
17973007 --X Sprites i cirklar. PB 86-2

99 Övrigt

01991002H --- Utskrift av månadskalender
01992001E --- Månkalender
07991005H --- Morse
09991003F --- Biorythm
10991007H --- Lotto
11993073 --X Tipsrad. PB 85-2
11997005 --- Slumpning av LOTTO
12993008 --- Toto
12993010 --- Tipsservice
12993011 --- Alkotest
13993021 -PX Etiketter
13993028 --X Biorytm II
14993032 --X Biorytm III
14993041 --- Resultatlista, slalom
14993049 --X Tips-system
15993078 -EX Automatisk uppringning
15993096 --- Dagräkning med almanacka
16993017 --- Barnvakten. PB 84-1

Spel från Thorn-EMI

Den minnesgode läsaren kanske minns att Thorn-EMI i England tog fram några spelmoduler för 99-an. I tidningen *Nittinian* nr 4-5/83 recenserades de av Peter Olin efter att ha visats på en mässa i Sollentuna. Han gav dem ett gott betyg och skrev att "Computer Wars och Submarine Commander utklassar alla TIs spel med god marginal". Tyvärr fick Thorn-EMI kalla fötter när TI kort därefter drog sig ur hemdatormarknaden och beslöt sig för att inte marknadsföra spelen.

En engelsk firma har i flera år försökt köpa rättigheterna till spelen. Under våren har de äntligen lyckats. De 3 spelen, de både ovannämnda och *River Rescue*, finns nu utgivna tillsammans på en diskett som kan laddas från Extended BASIC, Mini Memory och Editor/Assembler-modulerna. Priset är 35 pund. Firman som säljer heter Arcade Hardware och har adressen 211 Horton Road, Fallowfield, Manchester M147QE, England.

Senaste nytt

Den efterföljare till 99/4A som det ryktats om ända sen Texas lade ner produktionen av hemdatorn verkar nu vara närmare än tidigare. *Myarc* presenterade i början av juni en dator med namnet *Geneve* eller "model 9640 Family Computer". Enligt en annons ska maskinen ha en Texas 9995 processor, 640 K RAM, en 99/38 grafikprocessor och 128 K VDP RAM.

Det verkar som allt detta sitter på ett kort för expansionsboxen. Ett tangentbord av IBM PC-modell får man också med på köpet. Priset uppges till 495 dollar och det talas om leveransstart sista juli. Den nya datorn använder de gamla diskkontroll- och kommunikationskorten från antingen TI, Myarc eller CorComp. Det finns fortfarande massor av oklarheter om den nya datorn. Vi hoppas kunna återkomma i nästa nummer med utförligare information.