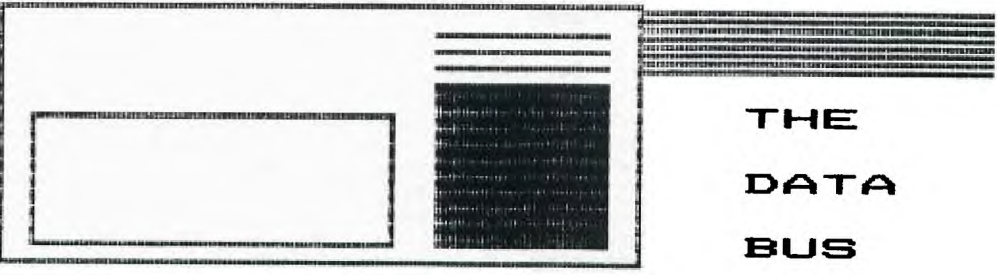


019 8704
Delaware

NEXT CHRISTIANA MEETING: APRIL 23

	<p>THE DATA BUS</p>	<p>ISSUE : VOL. 5 NO. 3 - - - A P R . 1 9 8 7</p>						
<p>THE DELAWARE VALLEY USERS GROUP DEDICATED TO THE TI AND COMPATIBLE HOME COMPUTER FAMILY</p> <p>P.O. BOX 6240 STANTON BRANCH, WILMINGTON DE 19804 4TH THURS. 6:30-9:30 P.M. CHRISTIANA MALL COMMUNITY ROOM</p> <hr/> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">KENT COUNTY, DE COURTHOUSE</td> <td style="width: 33%; text-align: center;"><CHAPTER></td> <td style="width: 33%;">DEPTFORD, NJ MUNICIPAL BLDG.</td> </tr> <tr> <td>2ND MONDAY 7:00-9:00 P.M.</td> <td style="text-align: center;"><MEETING></td> <td>3RD MONDAY 6:45-9:00 P.M.</td> </tr> </table>			KENT COUNTY, DE COURTHOUSE	<CHAPTER>	DEPTFORD, NJ MUNICIPAL BLDG.	2ND MONDAY 7:00-9:00 P.M.	<MEETING>	3RD MONDAY 6:45-9:00 P.M.
KENT COUNTY, DE COURTHOUSE	<CHAPTER>	DEPTFORD, NJ MUNICIPAL BLDG.						
2ND MONDAY 7:00-9:00 P.M.	<MEETING>	3RD MONDAY 6:45-9:00 P.M.						

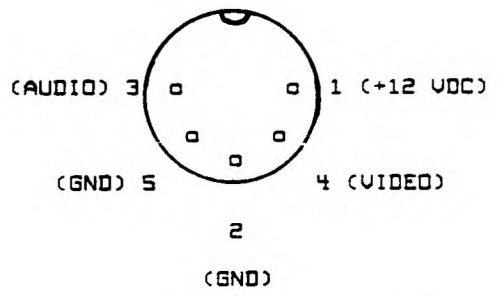
REM's on DISPLAYS
by K. Johnson, SFU 99ers

Video display units for computers come in two basic types: "monochrome" or "color". The screen is the face of a cathode ray tube (CRT). The most common example of a monochrome CRT display is the black and white TV. Other monochromes commonly used by computers are: green on black and amber on black.

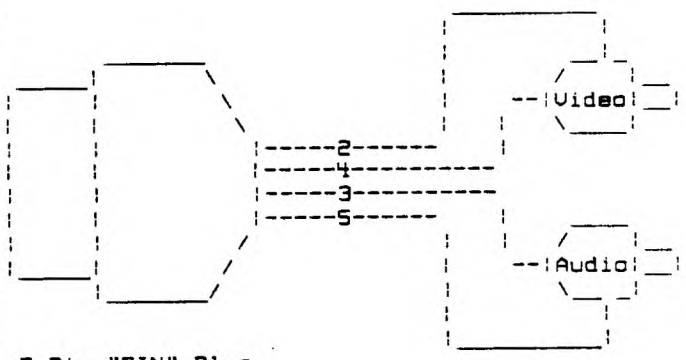
The video data from a home type computer to a monochrome (or color) display may be transmitted by one of two common methods: a radio frequency (RF) carrier modulated by the video signal or by direct connection of a "composite video" (CV) into the display. The RF method allows use of an ordinary TV as a display which must be set usually to either channel 3 or 4 depending on the RF Modulator unit. The TI 99 has a separate RF Modulator which gets the CV signal from the computer (along with operating power and the audio signal) and transmits it on the selected TV channel through the connection for the TV antenna. The quality of the display by this method is sometimes not too good because of the limited "resolution" of the normal TV system plus possible stray signal interferences from TV stations, Cable TV or even nearby video tape recorders (VTR) which may be operating on the same channel into some other TV.

Higher quality of graphic displays can be had by connecting a composite video "Monitor" directly to the CV and audio outputs of the computer. A CV Monitor is basically a TV set without RF receiving circuitry. Besides eliminating RF interferences, they are often built with "higher resolution" (more and smaller dots) than the normal TV. SONY (and more recently others) has for years built some models of their TVs with CV and audio input jacks so they may be used as CV Monitors. They were primarily for use with "Closed Circuit" TV or VTR's. Almost any CV Monitor of this type will work with the TI 99 and connected as shown:

RF Modulator Socket
(Left Rear of Computer)



CV Monitor



5 Pin "DIN" Plug
(Radio Shack)
(P/N 274-003)

CONTINUED ON PAGE 10

PAGE 2 - DELAWARE VALLEY USERS GROUP

DVUG EXECUTIVE COMMITTEE MEMBERS IN 1987

PRESIDENTTOM AUGUST
 VICE PRESIDENTJIM DAVIS
 SECRETARYTIM EVERS
 TREASURERTOM KLEIN
 SGT. AT ARMSJIM FOLZ
 DELMARVA CHAPTER CHRCHARLES BOWER
 SO. JERSEY CHAPTER CHRTONY DIFEBO

DVUG BULLETIN BOARDS

(302)322-3999 Anytime
 (609)429-7792 Monday-Thursday 3:00 PM-7:00 AM
 Friday 3:00 PM-Monday 7:00 AM
 (302)674-1449 6:00 PM-6:00 AM

For general information, you may contact
 TOM KLEIN Pa. (215)494-1372
 JACK SHATTUCK Del. (302)764-8619
 BUTCH FISHER N.J. (609)783-8276

Delaware Valley Users Group membership includes:
 library and software privileges, monthly DATABUS
 newsletter, plus other special benefits. Annual
 membership rates are: Family or Individual \$15;
 Student \$10; Newsletter only (beyond 75 mi) \$10.

TRANSMIT YOUR NEWSLETTER COPY ID: The Data Bus
 Editor --- Jim Folz, Telephone (302)995-6848, or
 use the DVUG mailing address shown on Page One.
 PLEASE SUBMIT NEWSLETTER ARTICLES FOR AN ISSUE
 BEFORE THE 2ND THURSDAY OF EACH MONTH.

An article appearing in The Data Bus may be
 reproduced for publication by another TI Users
 Group as long as acknowledgement is given to the
 sources as indicated. We encourage exchange
 newsletters; mail to DVUG business address shown
 on Page One.

DVUG ADVERTISING RATES FOR THE DATA BUS:
 1/4 page - \$ 5/issue, or \$ 45/12 issues
 1/2 page - \$ 8/issue, or \$ 75/12 issues
 Full page - \$15/issue, or \$125/12 issues

DELAWARE VALLEY USERS GROUP MEETINGS

Plenary meetings: Delaware's Christiana Mall on
 Rte. 7, at I-95 Exit 4-S, in the Community Room.
 Enter between J. C. Penney and Liberty Travel
 inside the Mall.

DELMARVA CHAPTER: Kent County Courthouse,
 Basement Conference Rm #25, Green & State Sts,
 Dover, De. Use the Green St. side entrance.

SOUTH JERSEY CHAPTER: Deptford Municipal
 Bldg, Cooper Ave. and Delsea Drive, (Rtes. 534 &
 47), in Gloucester County. Enter and park in
 rear of the building.

CONTENTS OF THE APRIL ISSUE OF THE DATA BUS:

REM's on DISPLAYS	Page 1
NOISE on The Data Bus	Page 2
Notes From T.I.C.O.F.F. 2	Page 3
BASIC/XB Programming Techniques	Page 4
Sorting	Pages 5-10
Plato	Page 10

NOISE on The Data Bus
 by Jim Folz

Regretfully, once again, this newsletter
 will be sent out later than was hoped. The
 delay this time was not caused by weather or
 late articles (although there were late
 articles). Instead, the weekend that I usually
 set aside for the newsletter was pre-empted by
 work needs. I have found that it takes at least
 three days plus a weekend to do the article
 gathering, formatting, and cutting-and-pasting
 when things go well. You will note that I did
 not include typing in the list. (An article
 like "Sorting" takes a long time with one
 finger). Some of you have volunteered to help
 with the typing. For this I thank you and I
 plan to take you up on it. I ask those of you
 that submit articles to REALLY try to have the
 articles to me by the second Thursday. The best
 way to get an article to me is by uploading to
 the Wilmington BBS. If we can do these things,
 then the newsletter production WILL go faster.
 This should result in faster delivery.

Let me assure you that the Deptford meeting
 date is foremost in my mind. The second
 Thursday deadline was set up based on the
 Deptford meeting date.

Now, on to the news ...

Unfortunately, the results of the Delmarva
 Chapter were received too late to be included in
 the March newsletter. Listed below are the
 Delmarva Chapter Officers for 1987.
 Congratulations and Good Luck!

President	Charles Bower
Vice President	Oscar Dawson
Secretary	Robert Edwards
Treasurer	Marian Bower
Software Librarian	James England

Apparently, the Christiana meeting place
 problem has been resolved. We have scheduled
 the Community Room at Christiana Mall for the
 fourth Thursday through October. Usually we
 have to move the November and December meeting
 dates due to Thanksgiving and Christmas. When
 we know those new dates I will place them here.

Don't forget that the Christiana group is
 raffling off a RAVE keyboard. Chances are \$3.00
 each and the drawing is to occur the night that
 the breakeven ticket is sold. The keyboard was
 picked up at T.I.C.O.F.F. and should be
 available at the next meeting. Don't miss out!

Thanks to Anne Dhein for some graphics
 software on instance conversion. The new
 software is in the software library. Contact
 Jack Shattuck if interested.

The Treasurer asks that chapter renewals be
 made at the chapter meetings. This reduces the
 amount of work in treasury adjusting each month.

Speaking of renewals, please check your
 renewal date. This is the time of year when
 many of the memberships expire.

NEWSLETTER EDITORS take note. Starting
 with the September 1986 DATABUS, articles
 appearing there have been placed in the DVUG
 library. Anyone interested in these or other
 DVUG holdings should contact us at the address
 on Page 1.



YOUR COMPUTER IS
IN THE MAIL ...



NOTES FROM TICOFF 2
by Jack Shattuck

In the week Myarc began shipping its Geneve 9640 computer to dealers, DVUG members at TICOFF 2 had a chance to see it running with Spad XIII, Not-Polyoptics' WWI air combat flight simulator. The program is written in pure Assembly Language but seemed like primitive animation when run on the II-99/4A compared with the action on Myarc's new system. Lou Phillips correctly noted that II game software would take on an entirely new look with the Geneve 9640. The cross-pattern keyboard cursor keys also seemed a great improvement.

Geneve's swan logo reportedly derives from the Swiss city, a fitting symbol of the Phoenix-like upgrade from what some thought an abandoned ugly duckling of a computer. (Who's an orphan, and who's abandoned it?!!)

As usual when Lou Phillips presents, he did quietly throw in an extra kicker. Finishing his address at TICOFF 2, he casually mentioned that the following week at Boston's "Fayuh", he'd be announcing Myarc's own bus box, which II owners lacking the Peripheral Expansion Box - and a NEW group of computer customers - could use to house the Geneve card as well as further offerings. No further details were immediately available.

The nearest dealers who are stocking Geneve and who "have (them) available for immediate shipment", according to a Myarc mailing, are Steve Citron's CITRONIC TECHNOLOGIES, 981 Townley Avenue, Union, NJ 07083 (201)686-5619 and Jeff Guide/Jim Horn of DISK ONLY SOFTWARE, P.O. Box 4170, Rockville, MD 20850 (301)369-1339. The prices may vary, probably between \$425 - \$475 in the minimum configuration (card and keyboard).

You'll get a manual and six software items with Geneve. These are Myarc's Disk Operating System (M-DOS), Advanced (i.e. Level 3.0) Basic, a Cartridge Saver to dump II cartridges to disk (over 30 modules can fit onto one DS/DD floppy), 4.21 PASCAL, and 80-column versions of Microsoft Multiplan and II-Writer, the latter allowing the retention of Editor and Formatter together.

To provide support for the Geneve community are two newly announced groups:

The MYARC SUPPORT GROUP asks folks to write to Don Iverson, 483 Valley Road, West Orange, NJ 07052 (Myarc's home state), or contact CIS EMAIL ID 74756, 413 Jay Holovacs.

J. Peter Hoddie, Barry Traver, Corson Wyman and others are requesting \$1 and your ideas for start-up of a Geneve newsletter. Tentatively to use the name GENIAL GENEVE, they can be reached by writing Genial Computerware, P.O. Box 183, in Grafton, MA 01519. That's Traver's name being redirected to New England, where Mr. Hoddie will presumably direct his graphic skills to a print outlet. There are a group of GENIAL programmers.

In case you hadn't heard, Hoddie has issued FONTWRITER II (see December's DATA BUS, p.3), to include Prowriter compatible systems. Asgard is the distributor at \$24.95, (301)345-2492.

The prolific Hoddie also had his Computer Shopper software first prize winner, XB:Bug, at TICOFF 2 for commercial sale at \$15, under that

Genial Computerware label. I got a copy, which provides features beyond XB Detective, for the debugging of Basic/XBasic programs. Either will serve you well (32K is needed), although the use of a screen menu in Detective may offer an extra convenience for some over the short 15-pg. "Bug" manual (half of which is filled with examples).

DVUG at TICOFF was prominently situated and exchanged contacts with numerous other groups of II users from out of town. Some local Jersey II owners who never heard of us before should be in touch with the Deptford meeting, as a result.

Norm Sellers got rearranged on the schedule for what I believe was the second straight year, but gave a strong demonstration of his A/L music preprocessor software. John Kelley's discussion of the II PLATO Beta program intrigued not a few persons; we sold eight sets of material, but the group most interested was the II Boston Computer Society contingent, who also took a copy of the protocols. It seems that the acquisition of this package is a major coup by our IIBBS committee.

Boston's group (BCS IIUG) provided Fairware for sale, i.e. first taking a suggested donation on the spot, using a form to send authors so the user would be registered for notification of any subsequent updates. That's not a bad idea, but at variance with the understanding that lets the user determine by trial whether or not a program is of benefit or worthy of such acknowledgment. They also were selling public domain disks at a \$3 cost. Other User Groups had a few programs or PD disks for sale, too.

I decided to get a little exotic and bought the BCS PD Disk #67, with three music programs. One, Six Ribbons, is a little minstrel tune, on the disk mostly as filler. The other two pieces are Wagnerian operatic music with an appropriate brooding graphic setting for each.

A more forboding mountain foreground, with ominous constantly shifting and darkening skies, is hard to imagine when you see the accompanying graphic display for Siegfried's funeral music in Die Gotterdammerung, Act III, Scene III.

A striking, yet distancing, reflective view of the Rhine is background for the Prelude to Tristan und Isolde. Such an unusual pair! They were composed by Ken Gilliland, whose address is included in the REM remarks. There's a note to the effect that the powerful Siegfried piece won a II Swap programming contest, with no further information provided; the composer is a resident of California. I've placed a copy of this disk, called "Opera", in DVUG's software library.

Another new product obtained was the Orphan Survivor's Handbook by Ron Albright. It helped overcome Art Byer's disappointment that editors across II land couldn't issue a "best of" 1986 newsletter on disk. Covering varied languages on the II plus II-Writer, telecom modes and a list of User Groups (incomplete, DVUG omitted among others), we were pleased to see three reprints from 1986 DATA BUS issues - a tribute to DVUG's newsletter relevance and readability. At the end of the publication was a form to send for update material as part of a "Users Network 99", being led by Los Angeles User Group VP Terrie Masters.

There are a series of II fests this time of year; Boston, Denver, Los Angeles, Ottawa, plus TICOFF, during April and May. If you can't make any others (airline low fare until May 20), save your money for the Triton College (in suburban Chicago) area faire in November.

BASIC/XB PROGRAMMING TECHNIQUES

With a large number of newer users joining our group, we have seen the demand for refresher and introductory BASIC programming classes. DVUG used to hold them in 1983, and the need is there again. Both Deptford and Dover users have moved to respond to that need, and Northern Delaware members will have their chance shortly.

Here are some techniques recalled, from the earlier years of THE DATA BUS, in 1983 and 1985 issues. How often have some of our experienced programmers used these lately?

"DISPLAY AT" IN BASIC (Vol.1, No. 8, Aug. 1983): Originally offered by Jack Shattuck

```
90 REM CALL CLEAR OPTIONAL
100 R=12
110 C=1
120 GOSUB 190
130 R=6
140 C=5
150 GOSUB 190
160 OATA HERE'S HOW TO PRINT
  IN BASIC
170 DATA WITHOUT SCROLLING U
  P
180 GOTO 180
190 READ MS
200 FOR I=1 TO LEN(MS)
210 CALL HCHAR(R,C+I+1,ASC(
  EG$(MS,I,1)))
220 NEXT I
230 RETURN
```

This listing offers a variation to a BASIC "print" command, which requires upward scrolling with each line printed. CALL HCHAR will print characters without that need, and this routine demonstrates its use for entire lines at a time.

The reference in line 210 (C+I+1) insures that text intended to start in the first column, as set forth in line 110 (C=1), actually begins in column 3, the usual starting place of TI text since columns 1-2 (and 31-32) may be off an edge of the screen for some TV's used as monitors.

If you REALLY want to start in columns 1 or 2, use C+I-1 or C+1 respectively, in line 210.

This subroutine prints a single line. To add additional lines, give new coordinates (for example, as shown in lines 130-140), then GOSUB again. The actual message segment (MS) is read from OATA statements, after which the program returned to the main listing, a CALL KEY pause, or whatever. Here line 180 is a freeze line that makes the program hold until you FCTN 4 <BREAK>.

DATA statements don't need quotes to print. CALL CLEAR has been omitted since one advantage usually sought from a slow but useful CALL HCHAR command is to print more data without disturbing the previous screen display.

After entering this program, RUN it without a CALL CLEAR statement to see what's meant. Next CALL CLEAR and RUN.

If you want to use a comma in what you'll be displaying, you'll have to enclose the OATA statement in quotation marks (like your usual PRINT command), since the comma is treated as a reserved character in a DATA statement. Normally quotation marks aren't needed for DATA display.

ANOTHER OLDIE BUT GOODIE
by Barry Boland

This tip appeared in Vol. 3, No. 3 (April, 1985):

big, Better, BEST, or, How to Gain Memory by INCREASING Program Size! (Requires XBASIC, 32K Expansion, Disk Drive)

This was downloaded in February, 1985, from our very own IIBBS. It's yet another example of how we can help each other between meetings. The problem here was switching memory areas, not an "oversized" program.

RE: PROGRAM PROBLEMS
FROM: JIMI JONES - WILMINGTON, DE
TO: ALL

HI EVERYONE. I'M TRYING TO GET A PROGRAM TO RUN ON POWERUP USING THAT FEATURE OF EXTENDED BASIC, WHEN I SELECT EXTENDED BASIC THE DISK ACTIVATES AND I GET AN I/O ERROR MESSAGE. THE TITLE OF THE PROGRAM IS LOAO.

(If you have a disk program called "OSK1.LOAO", it is automatically run when XBASIC is selected from the TI Title Screen. That feature is often employed to bring up main menus on XBasic disks. - Editor)

IF I TYPE IN THE "CALL FILES(1)" STATEMENT, IT WILL LOAO, AND APPEAR TO BE O.K. THE SIZE OF THE PROGRAM ACCORDING TO THE DISK MANAGER IS 50 (sectors).

IS THERE ANY WAY OF INCORPORATING A STATEMENT WITHIN THE PROGRAM (OR OTHERWISE) TO FIX THIS PROBLEM? I MAY BE ASKING TOO MUCH, I REALIZE. THIS PROGRAM WAS ORGINALLY ON CASSETTE AND I HAD THIS PROBLEM THERE TOO. ANY HELP YOU CAN GIVE WILL BE GREATLY APPRECIATED. THANKS MUCK!

** JIMI **

RE: Program Problems
FROM: Barry Boland
TO: Jimi Jones

Jimi, this may sound dumb, BUT it works!

What you want to do is ADD SOME SIZE to your program, to make it trip over into the Internal/Variable 254 format which uses the extra 32K (expansion) memory ...

What I did was take my disk catalog program and renumber it (say RES 30000,1).

Save it in MERGE format (SAVE DSK1.CAT, MERGE) and then MERGE it in with the program you want to fix (by loading the program to be fixed, then entering MERGE DSK1.CAT) ...

Then SAVE that (new combined program) back to the disk ...

It should now work with the LOAO name as you want, just leave an END statement (that is, have Line 29999 ENO) before the extra program.

(To show how things repeat themselves, the technique here was raised at the last Christiana meeting, and was also recalled at one of the workshop sessions at Calvary Episcopal Church in North Wilmington. Remember, the "dumb" question is usually the one you fail to ask! - Ed.)

Sorting
by Thomas Coppens

An important programming problem is the sorting of numbers or strings. This means ordering the numbers (strings) in ascending or descending order (ascending or descending alphabetical order). In this article series we will try to give you a number of methods for sorting which can be programmed in BASIC. We will also try to see how efficient (this means quick!) each method is. Therefore, we will always sort the same array of numbers and give the time needed. By the way, you can check the time for yourself!!

1. BUBBLE SORT

In this rather simple method, one goes through the rows of numbers, for example, from left to right. When two adjacent numbers are in the wrong order (first bigger than second when ordering in ascending order), they are swapped. This procedure is repeated until no more swapping has to be done.

The advantages are: simple, short program.
The disadvantages: much swapping, much 'loop'-work and slow.

Here follows the program:

```

10 REM BUBBLE SORT
20 DIM A(100)
30 RANDOMIZE (2)
100 REM SET UP RANDOM ARRAY
Y
110 FOR I=1 TO 100
120 A(I)=INT(RND*(99999))
130 NEXT I
135 PRINT "START"
140 REM START OF SORT
150 FOR I=2 TO 100
160 IF A(I)>A(I-1) THEN 200
165 REM SWAP THEM
170 X=A(I)
180 A(I)=A(I-1)
190 A(I-1)=X
195 F=1
200 NEXT I
205 REM WAS THERE A SWAP?
210 IF F=0 THEN 260
220 F=0
230 GOTO 150
250 REM END OF SORT
260 PRINT "END"
265 REM PRINT SORTED ARRAY
270 FOR I=1 TO 100
280 PRINT A(I)
290 NEXT I
300 END

```

```

10 REM INTEGER FLAG SORT
20 DIM A(100)
30 RANDOMIZE (2)
100 REM SET UP RANDOM ARRAY
Y
110 FOR I=1 TO 100
120 A(I)=INT(RND*(99999))
130 NEXT I
135 PRINT "START"
140 REM START OF SORT
150 N=100
160 R=N
170 F=0
180 FOR I=1 TO R-1

```

```

190 IF A(I)<A(I+1) THEN 215
195 X=A(I)
200 A(I)=A(I+1)
210 A(I+1)=X
214 F=1
215 NEXT I
220 IF F=0 THEN 260
225 R=F
230 GOTO 170
250 REM END OF SORT
260 PRINT "END"
265 REM PRINT SORTED ARRAY
270 FOR I=1 TO 100
280 PRINT A(I)
290 NEXT I
300 END

```

The approximate times for sorting 100 elements are:

Bubble Sort : 3'20"
Integer Flag Sort : 2'31"

2. SHAKER SORT

When doing a bubble sort by hand on paper to see what happens, one remarks that the biggest element is fast bubbling up (after one loop it is in its place). This is not the case, however, for the smallest element. This problem is taken care of in the so-called shaker sort. In fact, this is a bubble sort that goes both ways. One time the loop goes from left to right, the next time from right to left. This makes the name obvious. It may be clear that this can be combined with all the special features for the amelioration of the bubble sort.

```

100 REM SHAKER SORT
110 DIM A(100)
120 RANDOMIZE (2)
130 REM SET UP RANDOM ARRAY
Y
140 FOR I=1 TO 100
150 A(I)=INT(RND*(99999))
160 NEXT I
170 PRINT "START"
180 REM START OF SORT
190 N=100
200 L=1
210 R=N
220 F=0
230 FOR I=L TO R-1
240 IF A(I)<A(I+1) THEN 290
250 X=A(I)
260 A(I)=A(I+1)
270 A(I+1)=X
280 F=1
290 NEXT I
300 IF F=0 THEN 460
310 R=R-1
320 IF R=L THEN 460
330 F=0
340 FOR I=R TO L+1 STEP -1
350 IF A(I)>A(I-1) THEN 400
360 X=A(I)
370 A(I)=A(I-1)
380 A(I-1)=X
390 F=1
400 NEXT I
410 IF F=0 THEN 460
420 L=L+1
430 IF L=R THEN 460
440 GOTO 220
450 REM END OF SORT

```

```

460 PRINT "END"
470 REM PRINT SORTED ARRAY
480 FOR I=1 TO 100
490 PRINT A(I)
500 NEXT I
510 END
    
```

The time needed to sort the same 100 elements is 2'20".

3. SWAP SORT

This is a different kind of sort. We begin looking for the smallest element of the list. This element then comes in the place of the first element. Then we look for the second smallest element, which replaces the second element, and so on.
The program listing is:

```

10 REM SWAP SORT
20 DIM A(100)
30 RANDOMIZE (2)
100 REM SET UP RANDOM ARRAY
Y
110 FOR I=1 TO 100
120 A(I)=INT(RND*(99999))
130 NEXT I
135 PRINT "START"
140 REM START OF SORT
145 N=100
150 FOR I=1 TO N-1
155 R=I
160 FOR J=I+1 TO N
170 IF A(R)<A(J) THEN 190
180 R=J
190 NEXT J
195 IF R=I THEN 230
200 X=A(I)
210 A(I)=A(R)
220 A(R)=X
230 NEXT I
250 REM END OF SORT
260 PRINT "END"
265 REM PRINT SORTED ARRAY
270 FOR I=1 TO 100
280 PRINT A(I)
290 NEXT I
300 END
    
```

The time for sorting the 100 numbers is 1'05".

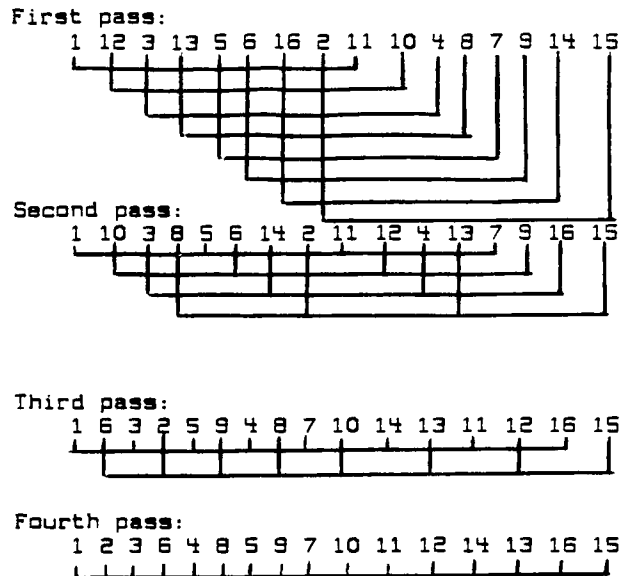
4. SHELL SORT

The shell sort is in fact an enhancement of the bubble sort. The idea behind this sort is to switch numbers over a larger distance first and then refine. Let's take an example:

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

Suppose we want to sort this list of 16 numbers. In this case, the sort begins with 8 groups of 2 numbers. These groups are sorted by one of the already mentioned methods. Then the list is divided in 4 groups of 4 numbers and each set is ordered. In this manner, the algorithm continues until we have one list of 16 numbers. After this list is sorted, the work is finished. Let's take a look at what happens in the

different passes of the sort:



Now the last sort takes place. This sort is a very good type of sort to use in programs. It is not too difficult to program in BASIC.
Here is a program of the shell sort:

```

10 REM SHELL SORT
20 DIM A(100)
30 RANDOMIZE (2)
100 REM SET UP RANDOM ARRAY
Y
110 FOR I=1 TO 100
120 A(I)=INT(RND*(99999))
130 NEXT I
135 PRINT "START"
140 REM START OF SORT
145 N=100
170 M=N
175 M=INT(M/2)
180 IF M=0 THEN 270
185 FOR SI=1 TO M
190 I=SI
195 J=SI+M
200 F=0
205 IF A(I)<A(J) THEN 235
210 F=1
215 X=A(I)
220 A(I)=A(J)
225 A(J)=X
235 I=J
240 J=J+M
245 IF J<N THEN 205
250 IF F=1 THEN 190
255 NEXT SI
260 GOTO 175
270 REM END OF SORT
275 PRINT "END"
276 REM PRINT SORTED ARRAY
280 FOR I=1 TO 100
285 PRINT A(I)
290 NEXT I
300 END
    
```

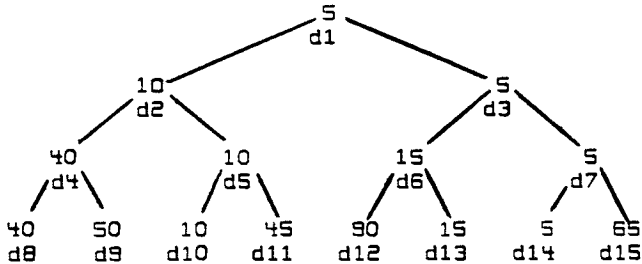
5. HEAP SORT

This is an entirely different kind of sort. Here the sorting is done by building binary

trees. A tree is an arrangement of elements as shown beneath. Take for example the numbers:

40 50 10 45 90 15 5 65

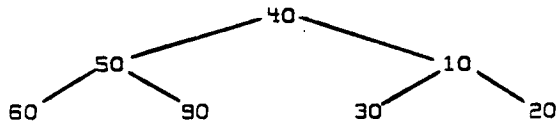
When ordering these numbers in a tree so that the smallest number is used as root, one needs N-1 comparisons.



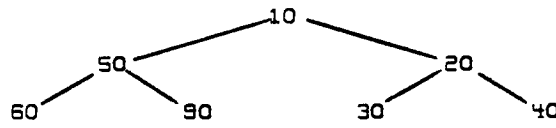
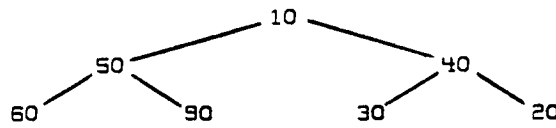
When such a binary tree has to be stored we have the following condition:

$$d(i) \leq d(2i) \text{ AND } d(i) \leq d(2i+1)$$

An ordering in that way is called a HEAP. Suppose now that the heap of the following figure has to take one more element (d1=40).



The new element is placed at the head of the heap. Then the element walks down the smaller elements and at the same time the small elements go upwards.



So the method of the heap sort is as follows:

- building of the heap
- working out the heap

A listing of the heap sort follows:

```

10 REM  HEAP SORT
20 DIM A(100)
30 RANDOMIZE (2)
40 REM  SET UP RANDOM ARRAY
50 FOR I=1 TO 100
60 A(I)=INT(RND*(99999))
70 NEXT I
80 PRINT "START"
90 REM  START OF SORT
100 N=100
110 L=INT(N/2)+1
120 M=N
130 IF L=1 THEN 170
140 L=L-1
150 X=A(L)
160 GOTO 210
170 X=A(M)

```

```

180 A(M)=A(1)
190 M=M-1
200 IF M=1 THEN 330
210 J=L
220 I=J
230 J=J*2
240 IF J=M THEN 280
250 IF J>M THEN 310
260 IF A(J)>A(J+1) THEN 280
270 J=J+1
280 IF X>A(J) THEN 310
290 A(I)=A(J)
300 GOTO 220
310 A(I)=X
320 GOTO 130
330 A(1)=X
331 PRINT "END"
340 FOR I=1 TO N
350 PRINT A(I)
360 NEXT I
370 END

```

6. QUICK SORT

This is probably the fastest sort. The procedure is as follows:

- choose an arbitrary element from the array to be sorted. For example: the element in the middle.
- search the array from the left and from the right to the middle. Do this until you find coming from the left an element larger than the chosen element and coming from the right an element smaller than the chosen element. Swap these elements and continue the procedure until the pointers meet each other. At that moment the array is split in two, where the elements in the left part are smaller than the elements in the right part.
- now sort each part of the array as mentioned above until the parts are formed by one element.

The algorithm is straightforward but the BASIC program is not that easy. For each partition the left and right ends have to be put in memory. This is done by building up a LIFO stack (this means a last in first out stack).

A program listing of the quick sort is given below:

```

10 REM  QUICK SORT
20 DIM A(100),ST(100,2)
30 RANDOMIZE (2)
40 REM  SET UP RANDOM ARRAY
50 FOR I=1 TO 100
60 A(I)=INT(RND*(99999))
70 NEXT I
80 PRINT "START"
90 REM  START OF SORT
100 N=100
110 L=1
120 R=N
130 I=0
140 X=A(INT((L+R)/2))
150 I=L
160 J=R
170 IF A(I)>X THEN 200
180 I=I+1
190 GOTO 170
200 IF A(J)<=X THEN 230
210 J=J-1
220 GOTO 200

```

1987														
JANUARY							FEBRUARY							
S	M	T	W	T	F	S	S	M	T	W	T	F	S	
				1	2	3	1	2	3	4	5	6	7	
4	5	6	7	8	9	10	8	9	10	11	12	13	14	
11	12	13	14	15	16	17	15	16	17	18	19	20	21	
18	19	20	21	22	23	24	22	23	24	25	26	27	28	
25	26	27	28	29	30	31								
MARCH							APRIL							
S	M	T	W	T	F	S	S	M	T	W	T	F	S	
1	2	3	4	5	6	7				1	2	3	4	
8	9	10	11	12	13	14	5	6	7	8	9	10	11	
15	16	17	18	19	20	21	12	13	14	15	16	17	18	
22	23	24	25	26	27	28	19	20	21	22	23	24	25	
29	30	31					26	27	28	29				
MAY							JUNE							
S	M	T	W	T	F	S	S	M	T	W	T	F	S	
				1	2		1	2	3	4	5	6		
3	4	5	6	7	8	9	7	8	9	10	11	12	13	
10	11	12	13	14	15	16	14	15	16	17	18	19	20	
17	18	19	20	21	22	23	21	22	23	24	25	26	27	
24	25	26	27	28	29	30	28	29	30					
31														
JULY							AUGUST							
S	M	T	W	T	F	S	S	M	T	W	T	F	S	
			1	2	3	4							1	
5	6	7	8	9	10	11	2	3	4	5	6	7	8	
12	13	14	15	16	17	18	9	10	11	12	13	14	15	
19	20	21	22	23	24	25	16	17	18	19	20	21	22	
26	27	28	29	30	31		23	24	25	26	27	28	29	
							30	31						
SEPTEMBER							OCTOBER							
S	M	T	W	T	F	S	S	M	T	W	T	F	S	
			1	2	3	4				1	2	3		
6	7	8	9	10	11	12	4	5	6	7	8	9	10	
13	14	15	16	17	18	19	11	12	13	14	15	16	17	
20	21	22	23	24	25	26	18	19	20	21	22	23	24	
27	28	29	30				25	26	27	28	29	30	31	
NOVEMBER							DECEMBER							
S	M	T	W	T	F	S	S	M	T	W	T	F	S	
1	2	3	4	5	6	7				1	2	3	4	5
8	9	10	11	12	13	14	6	7	8	9	10	11	12	
15	16	17	18	19	20	21	13	14	15	16	17	18	19	
22	23	24	25	26	27	28	20	21	22	23	24	25	26	
29	30						27	28	29	30	31			

REMEMBER -
 CHRISTIANA
 GROUP MEETS
 APRIL 23 (4TH
 THURSDAY)



```

230 IF A(I)<>A(J) THEN 270
240 IF I>=J THEN 270
250 I=I+1
260 GOTO 230
270 IF I>=J THEN 320
280 K=A(I)
290 A(I)=A(J)
300 A(J)=K
310 GOTO 170
320 I=I+1
330 J=J-1
340 IF I>=R THEN 380
350 T=T+1
360 ST(T,0)=I
370 ST(T,1)=R
380 R=J
390 IF L<R THEN 140
400 IF T=0 THEN 450
410 L=ST(T,0)
420 R=ST(T,1)
430 T=T-1
440 GOTO 140
450 PRINT "END"
460 FOR I=1 TO 100
470 PRINT A(I)
480 NEXT I
490 END
    
```

7. SORT DEMO

Now follows a program where all the sort methods we spoke about are included. It lets you choose the number of elements to be sorted and the type of sort you want to do. The word "START" appears on the screen, the sort begins and, when the sort is finished, the word "END" appears. So it is possible to time the sorting. We did this with the help of this program. The results are mentioned after the program listing.

```

5 CALL CLEAR
10 REM SORT DEMO
20 REM TISOFT (Belgian 99e
r club)
30 DIM A(400),B(400),ST(25,2
)
31 DATA "1. BUBBLESORT 1","2
","4. SHAKERSORT","5. SWAPSO
RT"
32 DATA "6. SHELLSORT","7. H
EAPSORT","8. QUICKSORT","9.
STOP PROGRAM"
35 RANDOMIZE (2)
36 FOR I=1 TO 9
37 READ WS(I)
38 NEXT I
40 CALL CLEAR
50 MS="*SORTDEMO*"
60 R=12
70 C=11
80 GOSUB 10000
90 MS="*****"
100 R=11
110 GOSUB 10000
120 R=13
130 GOSUB 10000
140 FOR DEL=1 TO 1000
150 NEXT DEL
160 CALL CLEAR
170 INPUT "HOW MANY NUMBERS?"
":G
180 IF INT(G)<>G THEN 160
190 IF G<10 THEN 160
200 IF G>500 THEN 160
    
```



```

210 CALL CLEAR
220 PRINT "JUST A MOMENT..."
230 PRINT "MAKING THE RANDOM
  ARRAY."
240 FOR I=1 TO G
250 B(I)=INT(RND*(99998))+1
260 NEXT I
270 CALL CLEAR
300 REM *CHOICE OF SORT*
305 C=3
310 FOR J=1 TO 9
320 MS=WS(J)
330 R=2*J+1
335 GOSUB 10000
340 NEXT J
540 MS="MAKE YOUR CHOICE (1-
  9)"
550 C=5
560 R=22
570 GOSUB 10000
580 CALL KEY(O,K,STAT)
590 IF STAT=0 THEN S80
600 IF K<49 THEN S80
610 IF K>57 THEN S80
620 ON K-48 GOSUB 1000,2000,
  3000,4000,5000,6000,7000,800
  0,9000
630 PRINT "END"
631 FOR I=1 TO G
632 CALL KEY(O,M,S)
633 IF S<>0 THEN 660
634 PRINT A(I)
636 NEXT I
640 FOR DEL=1 TO S00
650 NEXT DEL
660 CALL CLEAR
670 PRINT "PRESS ANY KEY FOR
  MENU"
680 CALL KEY(O,K,STAT)
690 IF STAT=0 THEN 680
700 CALL CLEAR
710 GOTO 300
1000 REM **BUBBLESORT 1**
1005 GOSUB 15000
1010 FOR I=2 TO N
1020 IF A(I)>A(I-1) THEN 10
  80
1040 X=A(I)
1050 A(I)=A(I-1)
1060 A(I-1)=X
1070 F=1
1080 NEXT I
1100 IF F=0 THEN 1130
1110 F=0
1120 GOTO 1010
1130 RETURN
2000 REM **BUBBLESORT 2**
2005 GOSUB 15000
2020 FOR I=2 TO N
2030 IF A(I)>A(I-1) THEN 20
  90
2050 X=A(I)
2060 A(I)=A(I-1)
2070 A(I-1)=X
2080 F=1
2090 NEXT I
2110 IF F=0 THEN 2150
2120 F=0
2130 N=N-1
2140 GOTO 2020
2150 RETURN
3000 REM **FLAGSORT**
3005 GOSUB 15000
3010 R=N
3020 F=0
3030 FOR I=1 TO R-1
3040 IF A(I)<A(I+1) THEN 30
  90
3050 X=A(I)
3060 A(I)=A(I+1)
3070 A(I+1)=X
3080 F=1
3090 NEXT I
3100 IF F=0 THEN 3130
3110 R=R-1
3120 GOTO 3020
3130 RETURN
4000 REM **SHAKERSORT**
4005 GOSUB 15000
4010 L=1
4020 R=N
4030 F=0
4040 FOR I=L TO R-1
4050 IF A(I)<A(I+1) THEN 41
  00
4060 X=A(I)
4070 A(I)=A(I+1)
4080 A(I+1)=X
4090 F=1
4100 NEXT I
4110 IF F=0 THEN 4260
4120 R=R-1
4130 IF R=L THEN 4260
4140 F=0
4150 FOR I=R TO L+1 STEP -1
4160 IF A(I)>A(I-1) THEN 42
  10
4170 X=A(I)
4180 A(I)=A(I-1)
4190 A(I-1)=X
4200 F=1
4210 NEXT I
4220 IF F=0 THEN 4260
4230 L=L+1
4240 IF L=R THEN 4260
4250 GOTO 4030
4260 RETURN
5000 REM **SWAPSORT**
5005 GOSUB 15000
5010 FOR I=1 TO N-1
5020 R=I
5030 FOR J=I+1 TO N
5040 IF A(R)<A(J) THEN 5060
5050 R=J
5060 NEXT J
5070 IF R=I THEN 5110
5080 X=A(I)
5090 A(I)=A(R)
5100 A(R)=X
5110 NEXT I
5120 RETURN
6000 REM **SHELLSORT**
6005 GOSUB 15000
6010 M=N
6020 M=INT(M/2)
6030 IF M=0 THEN 6190
6040 FOR S=1 TO M
6050 I=S
6060 J=S+M
6070 F=0
6080 IF A(I)<A(J) THEN 6130
6090 F=1
6100 X=A(I)
6110 A(I)=A(J)
6120 A(J)=X
6130 I=J
6140 J=J+M
6150 IF J<N THEN 6080
6160 IF F=1 THEN 6050
6170 NEXT S
6180 GOTO 6020
6190 RETURN
7000 REM **HEAPSORT**
7005 GOSUB 15000
7010 L=INT(N/2)+1
7020 M=N
7030 IF L=1 THEN 7070
7040 L=L-1
7050 X=A(L)
7060 GOTO 7110
7070 X=A(M)
7080 A(M)=A(L)
7090 M=M-1
7100 IF M=1 THEN 7230
7110 J=L
7120 I=J
7130 J=J*2
7140 IF J=M THEN 7180
7150 IF J>M THEN 7210
7160 IF A(J)>A(J+1) THEN 71
  80
7170 J=J+1
7180 IF X>A(J) THEN 7210
7190 A(I)=A(J)
7200 GOTO 7120
7210 A(I)=X
7220 GOTO 7030
7230 A(I)=X
7240 RETURN
8000 REM **QUICKSORT**
8005 GOSUB 15000
8010 L=1
8020 R=N
8030 T=0
8040 X=A(INT((L+R)/2))
8050 I=L
8060 J=R
8070 IF A(I)>X THEN 8100
8080 I=I+1
8090 GOTO 8070
8100 IF A(J)<X THEN 8130
8110 J=J-1
8120 GOTO 8100
8130 IF A(I)<>A(J) THEN 8170
8140 IF I>=J THEN 8170
8150 I=I+1
8160 GOTO 8130
8170 IF I>=J THEN 8220
8180 H=A(I)
8190 A(I)=A(J)
8200 A(J)=H
8210 GOTO 8070
8220 I=I+1
8230 J=J-1
8240 IF I>=R THEN 8280
8250 T=T+1
8260 ST(T,0)=I
8270 ST(T,1)=R
8280 R=J
8290 IF L<R THEN 8040
8300 IF T=0 THEN 8350
8310 L=ST(T,0)
8320 R=ST(T,1)
8330 T=T-1
8340 GOTO 8040
8350 RETURN
9000 REM *END*
9010 CALL CLEAR
9020 END
10000 REM *SCREEN DISPLAY*
10010 FOR I=1 TO LEN(MS)
10020 CALL HCHAR(R,C+I-1,ASC
  (SEG$(MS,I,1)))
10030 NEXT I
10040 RETURN
15000 REM *RENEW ARRAY*
15010 FOR I=1 TO G
15020 A(I)=B(I)
15030 NEXT I
15040 N=G
15050 CALL CLEAR

```

