

New SuperPIMx is 'do everything' computer program

Written By CHARLES D. HOUSE

SIDE A: contents of tape

- CLOAD"A" PART1/BAS, sort-code loader, common to all versions. Load and run this program first.
- CLOAD"B" CASSPIM/BAS, for 16-to-48K cassette tape systems.
- CLOAD"C" LEV3PIM/BAS, works faster than CASSPIM/BAS, because Level3 and BASIC3 have additional string-handling capability, as does disk Basic. BASIC3 or Level 3 are required, and must be resident in system.
- CLOAD"D" STRNGPIM/BAS, works with Stringy Floppy, the disk-speed wafer-storage system from Exatron, 3555 Ryder st., Santa Clara CA 95051.
- CLOAD"E" UTILPIM/BAS, version for JPC Products' TC-8 high-speed cassette full operating system.

SIDE B:

- CLOAD"F" DISKPIM/BAS, disk SuperPIMx (distributed on cassette)
- CLOAD"G" B17PIMX/BAS, version for use with software-based \$24.95 2100 bps cassette system from ABS Suppliers, also distributed by CIE. Speeds data save (and recovery) as much as 400 times Tandy rate, and greatly improves reliability.
- CLOAD"H" ZOOMPIM/BAS, for ZOOM 3.6 high-speed (3600 bps) tape interface from ZOOM! Box 3766, Nashua NH 03061.
- CLOAD"I" DISKTAPE/BAS, cassette module enabling disk user to also save file to tape, or load from tape. If you want this feature, merge DISKTAPE/BAS into DISKPIM/BAS. ZOOM, TC-8, B17 and Stringy users can use this module as a guide for either adding this feature to their system, or to add one of those systems as disk options.

Now, with one \$15.95 program from CIE, and no programming, your TRS-80 computer can help with many of the boring details of modern life, whether you be a businessman or student, professional or just citizen.

CIE believes this is the first major consumer breakthrough in software. Programs of this quality have not been available for less than 10 times the cost. SuperPIMx is to software what the TRS-80 was to computers—a price breakthrough.

With this one program, any TRS-80 owner can use his computer to do the following:

- Maintain checkbook balance,
- Keep a current list of department store charges,
- List personal inventory in case it is necessary for insurance purposes,
- Keep track of tax-deductible expenses, by category, for use in preparing tax forms,
- Direct mail, invitations, notices, cards to friends or members of club or for business purposes.
- Keep track of computer programs, music, books and articles.
- Use your "electronic file clerk" to organize research for a book or term paper, thesis, dissertation or article.

SuperPIMx is the electronic equivalent of an accountant's columnar pad. There are up to 20 columns, and a 48K system will allow in excess of 200 records, or to continue the simile to an accountant's pad, "entries".

TERMS USED IN DATABASE

Lest "entries" confuse you (with the key Enter), let us use the real words of database management: field, record, and file.

A field is what our accountant would call a "column" on his pad. Each field has its own name. Our accountant would write that name up at the top of the column.

A record is one of the horizontal lines on the accountant's pad. It includes one entry for every field (column) in the database.

A file is a group of records belonging to a database.

At CIE, we have a database of address records, more than 20,000 of them. This database is being converted to SuperPIMx because SuperPIMx is faster, easier to use, more flexible and versatile—more powerful than the expensive system we were using before.

Our old database had four fields: code gave us information about how the person's name got on the list, and a name line, address line and city-state-zip line.

With SuperPIMx, we are carving the record up into more fields. Besides code, we have first and last name, address, city, state and zip. We now can sort on any of those fields in far under ten seconds, where we once could sort only on zip, and the average file took 10 to 20 times that long. Now we can routinely do an alphabetical sort and duplicate-purge, then a zip sort and duplicate purge, all in a fraction of the time it once took us only to sort by zip.

LOWERCASE

One of the nicest features of the new database conversion, however, is going to be what we call pseudo lowercase. No longer will CIE News be coming with a label written in all capital letters. SuperPIMx will rewrite the database in caps and lowercase, a rather civilized touch, we think. Within the foreseeable future, we think we will be able to do word processing with this database. Think about that. Little CIE, with its five TRS-80 model 1s, thinks it soon could write 20,000 letters in a couple work days, nearly all automatically. We're already talking to a manufacturer about a demonstration, using a 2400 lpm, \$3950 printer.

WRITING TERM PAPER

While the ability to maintain a 20,000-plus-name address list indicates considerable power, let's consider what might be a more common application, writing an eighth-grade term paper on U.S. presidents.

Our eighth grader consults several references and decides he wants to list the presidents. He creates a database which includes the following: name, number, birth date, birth place, education, religion, married to... (and date), children, political party, previous biography, term of office, administration highlights, death date, death place, death cause.

Our student does not really know what he wants to do with this term paper. He needs to be able to assemble data and work with it, then bring it out one way and another.

He has come up with 15 fields, the things our accountant would call columnar headings. This is well within the limits of SuperPIMx, which holds us to no more than 20. (Should the user want more than 20 fields, he could simply change a DIM20 statement to DIM30 or some other number, within the limits of user memory).

UNCRAMPED STORAGE

Using most sequential database programs, the student would be required to define length of each of his 15 fields, and they could not total more than 255 characters (many programmers hold it to 240 as a safety margin).

As in most other areas of comparison, SuperPIMx is different. With SuperPIMx,

our student will be able to have variable-length fields, and any or all fields can be as long as 255 characters. Thus, with SuperPIMx, records can total 5,100 characters each!

Printout can be as a list, with the fields each on a different line, or columnar. Sorts are via machine language, and are very fast. Any field can be sorted. Thus our

As we gain experience with SuperPIMx, we will be developing a comprehensive user manual that goes beyond normal documentation supplied. Until then, we recommend the SCELBI publication: PIMS Personal Information Management System, \$11.95 from CIE and other booksellers. Both the book and the program are postpaid, but CA residents should add 6 per cent tax (\$12.67 and \$16.91, respectively).

Incidentally, SuperPIMx works with tape and/or disk systems 16-to-48K, as well as Stringy Floppy, Zoom and TC8 Poor Man's Floppy, and other systems.

student can print the list of presidents arranged via religious persuasion, or chronologically, alphabetically—even by wife's maiden name, if he wished.

As our student becomes more advanced, he can create data bases on each president, or on a more restricted subject, such as Indian affairs or Westward expansion.

SERIES OF ENHANCEMENTS

SuperPIMx is the basic "do everything" program that lets people use a computer for something utilitarian even if they do not know how to program. Of course there are limits to what can be put in even a "do everything" program. So there will be more advanced versions of SuperPIMx that will have more powerful features.

Advanced Labels will be a version that CIE uses to maintain its address lists. This will put an index on each data disk, and allow the program to print all the files on a disk without any operator intervention. Each additional disk will only require that the operator put the diskette in the drive, and press enter.

Advanced File Handling will be another extension. It will allow additional mathematical calculations. SuperPIMx now allows you to sum or sum and average any field. The calculator option will allow at least "four banger" math functions.

None of the advanced versions are yet complete, and are not yet ready for announcement. We do know that Advanced Labels will be \$24.95, and the other two will be \$5.95 each. These are only extensions of SuperPIMx, and the basic module is required! We will have more about SuperPIMx, and extensions, next month.

Meanwhile, deliveries have started on SuperPIMx, and we are anxious to hear how users have put it to good use. Let us know any features that should be built into it.

* SUPERPIMX 1.1 *

Changes and Improvements to SuperPimx 1.0

1. User will set MEMORY SIZE for all configurations. (See How to determine MEMORY SIZE) ... below.
2. PART 1 is now common to all configurations. The purpose of PART 1 is to load the Machine Language SORT code and, if applicable, KB DEBOUNCE.
3. MENU has 20 selections for a direct response to a user request. (See COMMAND TABLE) .. below.
4. DESCENDING as well as ASCENDING SORT capability has been added.
5. Files may now be MERGED to add to a File in Memory.
6. FIELD NAMES may now be added with a direct shift to the first RECORD so that data may be added to the new FIELD.
7. SEARCHES can now be conducted throughout the entire FILE as well as by individual FIELD.
8. FIELDS may now be ERASED as well as RECORDS. This feature allows greater flexibility for TABULAR printing.
9. FIELD NAMES, RECORDS and FIELDS can be REVISED.
10. SUMS will be calculated automatically on ALL numerical fields IF THE FIELD NAME is ended with a "#".
11. TABULAR printing will now allow PAGINATION, SINGLE or DOUBLE spacing, TOTALING of numerical fields that have been SUMMED prior to Tab printing.

HOW TO USE SUPERPIMX 1.1

1. Determine MEMORY SIZE and enter the value before loading and RUNNING the BASIC program titled "PART1/BAS". If you have difficulty in determining the correct MEMORY SIZE, go ahead and RUN "PART1/BAS". The program will TELL YOU THE CORRECT MEMORY SIZE you SHOULD have entered. You can then START OVER, enter the correct MEMORY SIZE and continue.

How to determine MEMORY SIZE:

If you are NOT using a HIGH MEMORY machine language program such as an RS-232 printer routine OR a program such as ZDATA, UTIL, B17, etc. then ASSIGN a value to ZT of 127 for 16K or 191 for 32K or 255 for 48K. NOW, multiply ZT times 256 and subtract 1. (subtact 57 if you will use KBFIX)

THE FORMULA: MEM SIZE=ZT*256-1 or ZT*256-57 for KBFIX

If you ARE using a HIGH MEMORY machine language program, the procedure is slightly different. DIVIDE the MEMORY SIZE value of the machine language program by 256 to obtain the value of ZT. For example, if you are using 'ZDATA' with a MEMORY SIZE of 65036, the formula is: ZT=INT(65036/256)-1. In this case, ZT=253. After you

have determined ZT, follow the FORMULA in the above paragraph to find MEM SIZE. ($ZT * 256 - 1$ or -57).

2. RUN "PART1/BAS" or CLOAD "PART1" if using cassette. This program has only one function and that is to load the machine language code for SORT and if applicable, KBFIX. The program will end when this function is completed.
3. LOAD or CLOAD the MAIN PROGRAM. (DISKPIMX, CASSPIMX, LEV3PIMX, etc.)
4. BEFORE YOU RUN THE PROGRAM . . . there are TWO things for you to do to make effective use of the program.

(1) It is MANDATORY that you check LINE 447 and EDIT if necessary, to change the value of DEFUSR= or POKE 16527, to the CORRECT VALUE of ZT. Otherwise, the SORT function will not operate. The values are set to 255 in the program you receive. Once changed and the program is saved, this step will NO longer be required.

(2) MEMORY MANAGEMENT: After you have checked Line 447 for the correct value, you should check your MEMORY by the direct command of PRINTMEM and ENTER. Now, check Line 104 to see the CLEAR value (set to 20000). A rule of thumb value should be roughly 70% of your MEMORY. If you are using a 16K capacity system, you will find that your Memory is inadequate to create files of more than 20 or so records with 6 to 7 Fields. However, MR. JAN STANCZAK of Strongsville, Ohio came up with a neat solution to this problem. (See LIMITED MEMORY SYSTEMS) below. Next, EDIT line 104 and set CLEAR to an amount compatible with your system and file structure. More comments on MEMORY MANAGEMENT: The DEFAULT value for RECORDS is automatically set to 100 and the DEFAULT value for FIELDS is set to 10. You are given an opportunity to change these values when you first RUN the program. The limiting factors are the amount of memory available, the size of your FIELDS and RECORDS and that Arch-Enemy of all TRS-80 systems, the GARBAGE COLLECTOR. You will encounter this insidious monster only occasionally if you will limit your FILES to 100 RECORDS or less and FIELDS to 10 or under. If your machine suddenly appears to stop and 'freeze' during a "SORT" or "SAVE" operation (most common), just go and enjoy a leisurely cup of coffee or tea and when you return in several minutes, the program will be functioning normally. Again, you can eliminate this problem by keeping your FILES relatively small. Computer Information Exchange can effectively manage a mailing list in excess of 20000 records by limiting Files to ZIPCODES of common areas. Most Files will contain less than 80 Records with 7 Fields per Record. Much more about address lists in the upcoming PIMx manual.

THE MAIN PROGRAM

CURSORS: While operating the program you will be asked to respond by pressing the 'ENTER' key or another key such as a '1' or an 'L' etc. When a SINGLE KEY is All that is required, you will see a small FLASHING CURSOR. You will see the large NON-FLASHING CURSOR when your KEY or DATA entry REQUIRES that you press the 'ENTER' key AFTER completing your response.

NOTE: DISK or LEVEL3 users may eliminate the 200+ Byte Solid Cursor routine at Lines 960 to 972 with short LINEINPUT and RETURN statement at Line 960:

```
960 F$=" ":PRINTCHR$(14);:LINEINPUTF$:PRINTCHR$(15);:RETURN
      However, most USERS seem to prefer the CURSOR.
```

THE MENU OR COMMAND TABLE:

The first thing to appear on the screen after you have indicated the the number of RECORDS and FIELDS that you intend to use will be the COMMAND TABLE.

COMMANDS ARE:

CREATE A FILE.....	1	CHANGE A RECORD.....	11
INPUT (LOAD) A FILE.....	2	CHANGE A FIELD NAME.....	12
OUTPUT (SAVE) A FILE.....	3	SORT IN ASCENDING ORDER...	13
MERGE A FILE.....	4	SORT IN DESCENDING ORDER..	14
LIST TO SCREEN.....	5	SUM NUMERICAL FIELDS.....	15
ADD RECORD(S) TO FILE.....	6	SELECT PSEUDO L/C.....	16
ADD A FIELD NAME.....	7	PRINT LABELS.....	17
SEARCH A FILE.....	8	PRINT TABULAR FORMAT.....	18
ERASE RECORDS.....	9	OTHER MODULE.....	19
ERASE FIELDS.....	10	END THE PROGRAM.....	20

PLEASE SELECT A NUMBER -

1. CREATE A FILE:

You are asked to Type the Name of your FIELDS and ENTER them, one by one. If you want the field data that you enter later to be stored in PSEUDO lower case (useful in printing LABELS) end the field name with "*", such as NAME*. Later, when you enter JOHN K MCDUFF in RECORD 1 FIELD 1, it will be stored so as to print John K McDuff. This will be done automatically without any shift key manipulation. If you do not end the FIELD NAME with a "*", you may still accomplish this feature by selecting 16 from the COMMAND TABLE. CAUTION: If you use this function, when you try to SEARCH a FILE, you will have to use the shift key (SHIFT for lower case letters, (NO-SHIFT for capitals) so the SEARCH routine can recognize the expression you are searching for.

For FIELDS that will contain NUMBERS that will later be SUMMED and that are of different lengths, END THE FIELD

NAME with a '*' such as AMOUNT*. (not necessary for ZIPCODES).

There is no need to refrain from using comma's or other punctuation symbols when entering data into FIELDS or when you are CREATING FIELD NAMES. The Field Name could be up to 255 characters in Length. Recommended length for neat formatting of both screen and printer is 1 to 20 characters. TO ESCAPE or quit, press 'ENTER' before entering a FIELD NAME when prompted to do so.

2. INPUT (LOAD) A FILE:

FILE previously SAVED by this program. When asked, type the FILENAME of the FILE you are about to LOAD. Its DIMENSIONS will not exceed those that you have DIMENSIONED when you answered the RECORDS ? and FIELDS ? questions upon first running the program. IF YOU EVER GET AN ERROR MESSAGE WHEN LOADING, MERGING, OR SAVING FILES, 'BREAK' and TYPE 'CLOSE' and ENTER. This will close the file that was opened. THEN TYPE GOTO120 and ENTER. IF you have not lost your variables, you will have the COMMAND TABLE appear on the screen and you can try to find the problem or try the COMMAND again. To ESCAPE when prompted to Enter a FILENAME, press the 'BREAK' key and type GOTO120 and 'ENTER'.

3. OUTPUT (SAVE) A FILE

Make sure your DISK or CASSETTE system is ready to accept the 'SAVE' and proceed as noted in paragraph 3. ESCAPE is the same procedure as shown above in Paragraph 2.

4. MERGE A FILE:

The File to be MERGED should be compatible with the FILE you have in Memory .. that is, the NAMES and NUMBERS of FIELDS should be the SAME. The FILE to be MERGED will start one RECORD higher than your number of RECORDS in Memory and it will rename the FIELDS as you MERGE. No problem if the FIELD NAMES are the same in both FILES. CAUTION: YOU MUST HAVE PREVIOUSLY DIMENSIONED YOUR FIELDS AND RECORDS when you first ran the program to allow for the higher number of records you will have in Memory. Example: If you answered the RECORDS question with 100 and you are MERGING a FILE of 56 RECORDS on top of a FILE of 52 RECORDS, the total of 108 RECORDS will exceed your DIM structure and you will get an ERROR message. If this happens, follow the procedure in Paragraph 2 above. ESCAPE is also the same as Par. 2

5. LIST TO SCREEN:

You will have an option of selecting 1 record, a range of records, or all records. The records will be shown on the screen with the Record Number, the Name of each Field followed by the DATA in each Field. You may continue to the next Record by 'ENTER' or ESCAPE and return to the MENU by striking the 'SPACE BAR'

6. ADD RECORD(S) TO FILE:

You will automatically go to this routine upon completing your FIELD NAMES and again, upon ADDING another Field as in COMMAND # 6 below. Here is where you enter the DATA into each FIELD that were defined in paragraph 1. Field Names will be listed as you are asked to enter the data into each Field. The number of characters are limited from 1 to 255. Again, as in Field Names, the program is designed to give you better screen and printer formatting and display if you limit your data in each Field to a maximum of 20 to 30 characters. However, if your data is such that long strings are necessary, the formatting could be changed by the user to meet this requirement. Long strings of data in each field could also use up Memory in a hurry. For example: If you have DIMensioned for 100 RECORDS of 10 FIELDS each and each Field contained 50 characters THEN $100 * 10 * 50 = 50,000$. (FAR BEYOND THE CAPACITY OF THE TRS-80). SELECTING this function from the COMMAND TABLE is usually done when you have established a FILE and wish to ADD more RECORDS. Be sure that you have DIMensioned enough RECORDS or you will get an ERROR message. In this case 'BREAK' and GOTO120. NORMAL ESCAPE to the MENU is made by pressing the 'ENTER' key before entering data into ANY Field of the new RECORD. To be a valid record, each Field must contain at least one character. If you want to avoid entering Data into some Fields, it is suggested that a "-" character be entered.

7. ADD A FIELD NAME:

Your present Field Names will be shown and you will asked to enter the NAME of the next Field to be added to each Record. Again, did you DIMension enough Fields to allow the addition of a new one? If not, you will be stopped with an ERROR. 'BREAK' and GOTO120 to continue. You will be required to 'RUN' again to RE-DIMENSION Fields. Be sure to SAVE the File first to avoid losing it. If you want to STOP or ESCAPE, press 'ENTER' when asked to enter the new Field Name. Immediately after you have ADDED a new Field Name, the program will shift to the ADD a RECORD function to allow you to enter the DATA for the new Field in each Record starting with Record 1.

8. SEARCH A FILE:

You will be shown a list of your Field Names and be asked to ENTER the number of the Field you believe the Data is located. IF YOU WANT TO SEARCH ALL FIELDS: Press the ENTER key. After entering the Field Number or pressing the ENTER key, the program will ask for the EXPRESSION you are searching for. The first time that the Expression is FOUND, you will see the entire Record printed on the Screen. This will continue until all Fields specified have been Searched. At the end of the

Search, you will be shown the number of times the expression was found and the number of the Record and the number of Field in which it was located. ESCAPE by using the 'BREAK' key and the GOTO120 entry.

9. ERASE A RECORD:

You are given a choice of ONE, a RANGE, or All RECORDS. CAUTION: Once erased, the Record(s) will be lost forever. SAVE the file first if you think you may need the Record later. You will see the routine in action on the screen as it shows you what Records it is eliminating and the process of re-numbering the Records in the FILE. ESCAPE with the 'BREAK' and GOTO120 entry.

10. ERASE FIELDS:

The Field Names are shown in order and you are asked to respond with a (Y/N) for this function. If you press a 'Y' key the program will immediately erase that Field Name and the Data in the Field. The process is shown on the screen as each Record is purged of the Field indicated and then the next Field Name is presented for your (Y/N) decision. ESCAPE by pressing the 'N' key for each Field in succession. This function is most useful for Tabular printing when you do not wish to show all of the Fields in each Record. REMEMBER TO SAVE YOUR FILE first if you do not want to permanently lose the erased Fields.

11. CHANGE A RECORD:

The program will respond with, "WHICH RECORD ?". After you enter the Record number, a list of all Fields in the Record are shown and then you are asked to enter the changed Data into each Field, one by one. If no change is required, the 'ENTER' key will leave the Field as is and proceed to the next Field in the Record for your perusal. ESCAPE by pressing ENTER as each Field is shown for your decision.

12. CHANGE A FIELD NAME:

This works the same way as the CHANGE A RECORD function above. The Field Names are really Record number "0" in your File. A\$(0,J). ESCAPE same as in Paragraph 11 above.

13. SORT IN ASCENDING ORDER:

You are shown a list of your Field Names and asked which Field to SORT by entering the number of the Field. DID YOU REMEMBER to CHECK Line 447 for the correct value of ZT before 'RUNNING' the Program? The EMMERT machine language sort is extremely fast. The only real delay is in the process of taking along all of the other fields in each record. This is done by a 5 to 10 second routine of indexing the sorted fields and then by assigning the new precedence numbers to each of the other fields in each Record. Here is a good place to encounter the

SNEAKY Garbage Collector. If your Files contain much over 100 records or if your Fields contain long strings of characters or if you did not CLEAR somewhat more String Space than needed, you may encounter a delay for this phenomenon. The computer will appear to 'Freeze-up' and repeated stiking of keys will have no effect. Never fear, just go for a short walk and when you return in a few minutes, the program will resume. ESCAPE (if before the Garbage Man) is by the 'BREAK' and GOTO120 routine.

14. SORT IN DESCENDING ORDER:
All of the comments in Paragraph 13, above apply. The routine may take 2-3 seconds longer as an additional array is used to change the numbers from low to high.
15. SUM NUMERICAL FIELDS:
The program will ask for the Range of Records or All Records to be summed. If you ended the Numerical Field NAMES with the "*" symbol, all of the pertinent fields will be summed for the Records that you indicated. If none of your Field Names were ended with the "*", then you will get an immediate <ENTER> invitation. Jus press ENTER and you will be back at the MENU. ESCAPE by the 'BREAK' and GOTO120 routine if you are stopped for some unknown reason.
16. SELECT PSEUDO L/C:
For some applications such as LABELS, this is a convenient option. The program will automatically store the first letter of each word in upper-case and the other letters in lower-case. If it encounters an MCBEAN or other MC-name it will store the word as McBean. The same is true for RR (abbreviation for rural-route). Both letters will be capitalized. You could have accomplished this same function by ending the Field Name with a "*". However, here you may have more versatility by using the function only as needed. The cost of using this option here instead of the Field Name method is that it seems to take more time. You are asked to indicate if you want this option by a <Y/N> response as you are stepped through the Field Names, one by one. A 'Y' key for the selected Fields will cause the program to change the Field in the Records you select by the 'ONE', 'RANGE' or 'ALL' option to Pseudo L/C. You will see the action taking place on the screen as the record and field numbers appear as each record is processed. ESCAPE by the 'BREAK' and GOTO120 method.
17. PRINT LABELS:
You are first asked if you want to ALIGN LABELS <Y/N>. A 'Y' response will print 5 lines of '*' with the 6th line blank. You will come back to the same question as many times as you press 'Y'. The 'N' key bypasses the align-

ment routine and asked you if you want 'MORE THAN ONE FIELD PER LINE (Y/N). A 'N' response will show you your Field Names and ask you to indicate the FIELD NUMBER you want printed on each Line. (up to 5 allowed). You could change this to more lines by editing this part of the program. After your LABEL format is set up, you may choose ONE, a RANGE or ALL RECORDS to print your Labels. The 'Y' response allows you to indicate up to 3 Fields to be printed on each of your 5 lines to a Label. To leave a Line Blank, just press the 'ENTER' key when asked for the Number of Fields you want on a Line. ESCAPE is via the 'BREAK' and GOTO120 method.

18. PRINT TABULAR FORMAT:

The prints a message asking you 'STANBY TO COMPUTE FIELD LENGTHS' while it computes the Format for printing all of your Fields with one space between Fields. This includes the Field Names which will be printed across the page with a line below. After completing the computation, you are shown where the 'TABS' will occur and the total number of columns you will be using. The computation only takes place the first time you select this function. You are also given an option to change the spacing from 1 space between Fields to 2 thru 9. You make experiment here freely because you can change these numbers as often as you like until you press the 'ENTER' key instead of a number. press ENTER when you are satisfied with the the spacing. Now you asked to indicate Single or Double Spacing between LINES and then to indicate the number of Lines you want on a Page for automatic TOP of FORM as you print your records. Next, you are given the option of printing a 3 Line Title before the File is printed. An ENTER key will cause the Line to be BLANK. After the 3rd Line of the Title is Printed, the File printout will begin. When the Records you have selected to be printed have been completed, the program will ask if You want TOTALS to be printed (where some of your fields are numerical) by the (Y/N) option. An 'N' will draw a double line of "=" at the end of the file. A 'Y' will draw a single line of "-" and show a dash under the Alpha fields with the sum printed under the Numeric Fields. Then the double line is drawn and you are returned to the MENU.

IMPORTANT: THE PROGRAM IS CONFIGURED TO PRINT THE TITLE LINES IN THE CENTER OF THE PAGE AND TO PRINT THE FILE IN LARGE CONSENSED PRINT. (130 LINES ACROSS AN 80 COLUMN PAGE). PROGRAM LINES 730, 732 AND 734 CONTAIN THE CHR\$() NUMBERS FOR THE LINE PRINTER VI YOU MAY HAVE TO CHANGE THESE VALUES TO CONFORM TO THE PRINTER YOU ARE USING.

19. OTHER MODULE:

At the present time, this selection does nothing but asks you to return to the MENU with 'ENTER' after

printing a message that this option "IS RESERVED FOR LATER USE". Several more modules are planned for incorporation into SUPERPIMX. Watch for them.

20. END THE PROGRAM:

Asks if you want to "SAVE YOUR FILE FIRST". If 'Y' you will go to the SAVE A FILE FUNCTION, otherwise the the program will CLEAR50, END and return to BASIC.

LIMITED MEMORY SYSTEMS:

16K Users and often Disk Based 32K systems users will find that their Memory Requirements are not adequate to use this program effectively because of a lack of RAM Memory.

Mr. Jan Stanczak of Strongsville Ohio has not found this to be much of a problem even though he has a 16K Level II TRS-80.

Mr. Stanczak found that by splitting the program into TWO parts he could do everything he wanted. Part 1, called his 'INPUT' program, was made up of only those features he wanted in order to Create, List, Sort, Search, Add, Change, and Save Files. Part 2, called the 'OUTPUT' program consists of the List, Search, Erase, Sort, Labels, Tabular, Load and Merge functions. He elected to forego the Pseudo-L/C function completely. By this unique solution, Mr. Stanczak found that he was able to effectively use the program to meet his needs.

The user could easily change the program to meet an individual need by finding deleting the portions of the program to be eliminated and deleting where necessary. The beginning lines of each function of the command table are listed in LINE 130. In most cases, the ending line of the function contains the instruction, "GOTO120". The Menu could be left as is, if a GOTO120 were entered on the beginning Line of a function to be deleted and the remaining Lines of that particular function deleted.

PIMX ERRATA:

Line 65 got clobbered in early releases of SuperPIMx 1.1, and if the user followed instructions given on the screen after loading of the sort code, the sort did not work.

Line 65 of the loader program, PART1/BAS, should read:

```

65 PRINT:
    PRINT:
    PRINT*AFTER YOU LOAD OR CLOAD THE MAIN
      PROGRAM...
    MAKE SURE THAT LINE 447 READS (IF DISK):
      DEFUSR=";ZT*256-65536
    (IF NONL-DISK):
      POKE 16526,0: POKE 16527,";ZT

```

NOTE: ZT times 256 is the position in memory where the sort starts (the sort-enter address). In disk you go to a define-user address. In non-disk ZT is poked into memory at 16527.

To correct this error, edit line 65, and hit the X key, which prints the line and puts you in the insert mode. Next key in the following:

```
*256-65536
```

So, the incomplete line reads:

```

65 PRINT: PRINT: PRINT *AFTER YOU LOAD OR CLOAD THE MAIN
PROGRAM... MAKE SURE THAT LINE 447 READS (IF DISK):
  DEFUSR=";ZT
65 PRINT: PRINT: PRINT *AFTER YOU LOAD OR CLOAD THE MAIN
PROGRAM... MAKE SURE THAT LINE 447 READS (IF
DISK):DEFUSR=";ZT*256-65536

```