Volume 6, Issue 4 April, 1984
$1.50

TRS-80® Microcomputer News

- Communications
- Databases

Fort Worth Scene

This is the month we feature some articles and programs on communications and databases. The database items range from the sophisticated Profile 16 to a tape based program for Genealogical records on the Color Computer. In addition to our regular communication contributors—CompuServe, AgriStar, and Al and Dru Simon—we found a couple of reader contributed programs which were particularly pertinent. There's the AutoDial program for the Modem II, a formatter for interface with your line printer, and a hint on using Model III/4 VideoTex Plus.

By the way, although they don't mention it in their article, the blizzard in January nearly caused us to miss the April edition of Communications Corner. From our talk with Dru, we gather they had to start from almost scratch to get online again! Our thanks to both Al and Dru for their dedication and continued contribution.

OUR PROGRAMS/YOUR TYPING—MAKING A MATCH

Honest, folks! Our goal is to provide you with easy to read programs. So, we format accordingly and in the process seem to cause some confusion for a few readers.

Program lines are printed with extra spaces and, due to space requirements, are divided into segments so that parts of a single program line may appear on multiple print lines. When you are entering these programs on your system, omit all unnecessary, extraneous spaces. For example a program line which appears as follows:

```
5Ø A+B=C
    : PRINT C
    : GOTO 3Ø
```

should be entered omitting all extra spaces. When you type the line it should look something like this:

```
5Ø A+B=C:PRINT C:GOTO 3Ø
```

We hedge a little there because, to be absolutely certain of the proper format for your system, you should read the BASIC section of your manual. Generally, the only required spaces (or spaces that you will want to be sure to maintain) are those that appear within quotes. Since the spacing requirements for BASIC may vary slightly from computer to computer, your best source of information on spacing is your system manual.

MODEL 2000

We are all pretty excited about the terrific Tandy Model 2000 and are delighted to be able to provide a program this month that was written specifically for this great new machine.

Thanks to a dedicated hacker, Earl Bollinger, we have a Model 2000 utility program, written in Pascal. And Earl is a sharing kind of fellow; you will be seeing more of his programming efforts in coming months.

OS9 COMMUNICATIONS

In case you hadn't noticed, Earl has also been generous with information regarding OS9. We have published several of his OS9 articles in recent months; this month's OS9 contribution deals with communication via Device Path Descriptor Modules. Earl explains these modules and provides a exemplary assembly language program.

ETC.

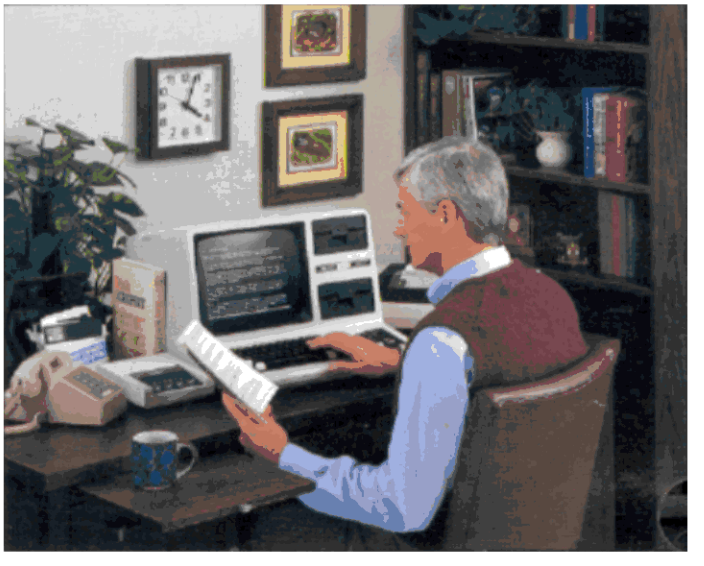
There's much more for you to discover. In addition to the articles, we've tried to provide a variety of programs so that there is something for every system.

Speaking of all those reader contributed programs, if you've sent in a program and have yet to see it in print, don't think it can't still happen. All contributions are maintained in our active files for an extended length of time and may be selected momentarily for publication.

1984 NATIONAL COMPUTER CAMPS

If your youngsters haven't signed up for National Computer Camp, now is the time. Campers may sign up for one or more weeks during June, July, and August. The camps are for boys and girls, ages 9-18, of all levels of computer experience, including no computer experience at all. The locations for the 1984 camps are: Simsbury, Connecticut; Atlanta, Georgia; St. Louis, Missouri; Cleveland, Ohio; and Portland, Oregon.

For further information contact Michael Zabinski, Ph.D. at (203) 795-9667, or write to National Computer Camps, Box 585, Orange, Ct, 06477.



Front Cover: home den with Model 4, Modem II, DMP200, and various communication, database, and other software.

TRS-80 Microcomputer News
Volume 6 Issue 4
APRIL 1984

TRS-80 Microcomputer News is published monthly by Radio Shack, a division of Tandy Corporation. A single six month subscription is available free to purchasers of new full size TRS-80 Microcomputer systems with addresses in the United States, Puerto Rico, Canada and APO or FPO addresses. Certain smaller TRS-80 Microcomputers will not include this free subscription. Subscriptions to other addresses are not available.

The subscription rate for renewals and other interested persons with U.S., APO or FPO addresses is twelve dollars ($12.00) per year, check or money order. Single copies of the Microcomputer News may be purchased from Radio Shack Computer Centers or Computer Departments for $1.50 suggested retail each.

The subscription rate for renewals and other interested persons with Canadian addresses is Fifteen dollars ($15.00) per year, check or money order in U.S. funds. All correspondence related to subscriptions should be sent to: Microcomputer News, P.O. Box 2910, Fort Worth, Texas 76113-2910.

Retail Prices in this newsletter may vary at individual stores and dealers. The company cannot be liable for pictorial and typographical inaccuracies.

Back issues of Microcomputer News prior to January, 1981 are available through your local Radio Shack store as stock number 26-2115 (Suggested Retail Price $4.95 for the set). Back issues of 1981 copies are available as stock number 26-2240 (Suggested Retail Price $9.95 for the set). The 1982 back issues copies are available as stock number 26-2241 (Suggested Retail Price $12.95 for the set).

The TRS-80 Newsletter welcomes the receipt of computer programs, or other material which you would like to make available to users of TRS-80 Microcomputer systems. In order for us to reprint your submission, you must specifically request that your material be considered for reprinting in the newsletter and provide no notice that you retain copyrights or other exclusive rights in the material. This assures that our readers may be permitted to recopy and use your material without creating any legal hassles.

Material for publication should be submitted on magnetic media (tape, disk, or CompuServe). If you submit material on tape or disk, and it is accepted for publication, we will send you two cassettes or diskettes for each one you sent us. Cassettes will come from our box of mixed blank cassettes. If you submit material on CompuServe, and we think we may use the material, we will extend your Microcomputer News subscription by six months for each article accepted. If you are submitting material over CompuServe, please include your name and address or your subscription number so we can find you. If the material is very short, send it to us in E-Mail. If you have more than a few lines, you need to place the material in the ACCESS area of CompuServe and then let us know it is there by leaving a message on E-Mail.

Material may be submitted by mail to P.O. Box 2910, Fort Worth, Texas 76113-2910, or through CompuServe. The Microcomputer News' CompuServe user ID number is 70007,535.

Programs published in the Microcomputer News are provided as is, for your information. While we make reasonable efforts to ensure that the programs we publish here work as specified, Radio Shack can not assume any liability for the accuracy either of the programs themselves or of the results provided by the programs.

Further, while Microcomputer News is a product of Radio Shack, the programs and much of the information published here are not Radio Shack products, and as such can not be supported by our Computer Customer Service group. If you have questions about a program in the Microcomputer News, your first option is to write directly to the author of the program. When possible, we are now including author's addresses to facilitate communications. If the address is not published, or if you are not happy with the response you get, please write us here at Microcomputer News. We will try (given the limited size of our staff) to find an answer to your question and, in many cases, will publish the answer in an up-coming issue of Microcomputer News.

TRS-80® Microcomputer News

Contents:

Prices shown in TRS-80 MICROCOMPUTER NEWS are in U.S. Funds.

Profile 16: Processing Tables and Dummy Fields

The Small Computer Company
P.O. Box 2910
Fort Worth, TX 76113-2910
By Ivan Sygoda, Director, Pentacle

You should have a look at Profile 16, which you can find at Radio Shack, even if you don't (yet) own a Model 16. It's the data base management system you've been dreaming about, with capabilities and features that will make your mouth water. It comes with an extensive user manual; however, don't be intimidated.

Profile 16 is as easy to use as Profile Plus and Profile III Plus because it has the same outward structure—program modules to create and then run screens, reports, labels, forms and Scripsit or Multiplan spinoff files. In many ways, it's even easier to use than its TRSDOS-based cousins because it's more flexible. You have much more leeway in designing key and data segments, for instance, and it's a lot easier to make changes after data has been entered.

Yet, at the same time, this flexibility can be a source of confusion when you first set up a data base, since so many choices and options are now available to you. However, it becomes quite manageable when you break things down into their smallest elements. This is the approach we'll take as we investigate Profile 16 during the coming months. Along the way, Profile Plus and Profile III Plus users will discover some ideas they can adapt for their own data files. After Profile 16 gets a solid launching, I'll once again devote a fair share of space to the Model II and III versions.

TOOLS TO BUILD WITH

The processing tables are your principal tools for making Profile 16 do exactly what you want it to do. These tables are descendents of the Profile Plus and Profile III Plus math tables, but they are much more powerful—as you'll see, they can do almost anything except make coffee.

If your application requires math, you'll be pleased with the tables' greatly increased sophistication. First of all, nested parentheses are supported, as well as minimum, maximum and averaging functions. Best of all, the tables let you do conditional processing. This means, for instance, that you can direct the program to perform one or more (or none) of a set of computations if certain criteria are met. The uses of this capability are almost unlimited — automatic calculation of sales taxes which vary by city or by state, sliding commission scales, computing bonuses or rebates, etc.

This month I'll focus on the ways you can manipulate text fields (character strings) using the tables. To illustrate, I'll use a relatively simple mailing list file I call "maillist." (It's a convention in TRS-XENIX to use lower case for file names and commands.) Figure 1 lists the fields I defined for this data base, along with their lengths and edit types. Consult last month's article for a brief explanation of these edits. The only new one is "ho," which is a user-defined edit I devised to restrict field 8 to "H" or "O." You'll see why this is important a little further on.

```
Jan  8 15:11 1984    File Name: maillist                          Page 1

      Number    -------------Field Heading--------------    Len   --Type--

key segment:
          1     Marker code                                    4   allup
          2     Category code                                  4   allup
          3     Mailing zip code                              10   zip
          4     Last name                                     15   *
          5     First name                                    12   *
          6     Organization                                  35   *
          7     Affiliation code                               3   allup
          8     Send mail to Home or Office? (H/O)             1   ho
          9     Reserve 1                                      0   *
         10     Reserve 2                                      0   *

data segment:
         11     Honorific                                     10   *
         12     Title                                         15   *
         13     Department                                    20   *
         14     Ofc./street address                           25   *
         15     Ofc./city                                     20   *
         16     Ofc./state                                     2   allup
         17     Ofc./zip                                      10   zip
         18     Ofc./phone                                    14   phone
         19     Home/street address                           25   *
         20     Home/city                                     20   *
         21     Home/state                                     2   allup
         22     Home/zip                                      10   zip
         23     Home/phone                                    14   phone
         24     Source code                                    3   allup
         25     Salutation: Dear...                           30   *
         26     Note                                          60   *

  Key segment record length:   84
Data segment record length:  280
```

Figure 1: The maillist fields

THREE OF A KIND

There are three different kinds of processing tables: automatic, input, and output. Each data base may have one automatic processing table, one input table, and any number of output tables. The output table takes its name from the output formats (reports, labels, forms, spinoff files and batch processing procedures) with which they are associated, and you can have as many different formats as you need.

All three tables look identical and are created according to the same structure and syntax rules. Furthermore, they all perform operations on the data contained in your fields. The difference between them is where and when they function. The automatic processing table operates before a requested record is displayed on the screen or accessed for a report form or other output. The input processing table does its thing at the moment you record new or updated data by pressing ⟨ESC⟩ ⟨ESC⟩. The output processing tables go into action as output is generated. What follows is an example of each.

```
Jan .8 21:35 1984    File Name: maillist                      Page 1
                     Processing: Automatic

  1 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If: 11 NE ""
      Then: NM(39,*)=11<5<4
  2 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If: 11 EQ ""
      Then: NM(39,*)=5<4
  3 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If: 12 NE "" AND 13 NE ""
      Then: TD(37,*)=12(","<13
  4 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If: 12 EQ "" OR 13 EQ ""
      Then: TD(37,*)=12(13
  5 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If:
      Then: CZ(33,*)=15(","<16<17
  6 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If:
      Then: HZ(33,*)=20(","<21<22
  7 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If:
      Then: SL(29,*)=25(","
```

Figure 2. The maillist automatic processing table.

```
Jan  4 11:49 1984    File Name: maillist              Screen #2    Page 1

*Pentacle maillist: Code entry
 !NM                                            Office phone:  !18          /
 !TD                                              Home phone:  !23          /
 !6                                  /        !19                         /
 !14                        /                 !HZ
 !CZ                                          Dear !SL
------------------------------------------------------------------------------
 Note: *26
                          Category code                  Marker:       S'rce: !24/
                              *2  /                           *1  /    Affil: !7 /
 .................................... Reserved   ..............
                               ..............   Newsletter ...
 Profession                     Panel      VIP    ..............     Created
Art.Dir./Chor.  Reg/State arts Dance/NEA  Alist  Friends       ..   on !@CD     /
Dancer          Wierd sponsor  Inter/NEA  Blist  Audition      ..   by !@CB
Manager         YAgent         NYSCA      *Pent  Chor.proj.    ..
Board member    ZSupplier      Riverside         ..............     Updated
Service prof.   EDance studio  CAPS              Pentacle mail.     on !@UD
Fndn. exec.                    Auditor           ..............     by !@UB
Corp. exec.     Other                             Reserved
```

Figure 4. Maillist screen 2, showing "processed" entries.

AUTOMATIC PROCESSING

Figure 2 shows the automatic processing table for maillist. It also illustrates the use of "dummy" fields and some of Profile 16's considerable string handling capabilities.

Each processing element (there are a possible 200 of them on each table) has three parts: an optional "label," an optional "condition" line titled "If" and an "action" line titled "Then." Labels are used to redirect the flow of processing (which normally runs down the table from element 1 to the end), using GOTO or GOSUB statements, as in BASIC programming. (In BASIC, GOTO and GOSUB point to line numbers; in Profile 16, they point to labels.) See the example below, on the output processing table in figure 7.

In this maillist application, I use automatic processing mainly for cosmetic purposes — to organize the fields entered on screen 1 (figure 3) so that they appear on screen 2 (figure 4) as they might if they were entered manually. Field 11 holds "Ms." or "Mr." Field 5 is the first name and field 4 is the last name. If these fields were placed on the screen one after the other, the result would be something like this:

Mr.   Ivan   Sygoda

when what I really want is this:

Mr. Ivan Sygoda

which looks better and is easier to read.

The operation generated by the action ("then") line in automatic processing element 1 can be described in English as follows: 1) Take field 11 (Ms., Mr., etc.) and strip away any unused spaces—Mr., for instance, occupies only three of the allotted ten characters; 2) leave the usual single space between words; 3) then put field 5 (the first name), also stripped of trailing blanks after that; and 4) leave another space; 5)

```
Jan  4 11:48 1984    File Name: maillist              Screen #1    Page 1

 *Pentacle maillist: Data entry
..............................................................................
     Ms./Mr.:  *11       /                                   Source:  *24/
  First name:  *5         /                             Affiliation:  *7 /
   Last name:  *4            /                             Category:  *2  /
       Title:  *12           /                               Marker:  *1  /
  Department:  *13                /
Organization:  *6                               /
     Address:  *14                      /
        City:  *15                /   State:  *16   Zip:  *17        /
Office phone:  *18         /
..............................................................................
Home address:  *19                      /
        City:  *20                /   State:  *21   Zip:  *22        /
  Home phone:  *23         /
..............................................................................
  Salutation:  Dear *25                         /   Mail to H/O?  *8
        Note:  *26                                                         /
..............................................................................
Record!@RN    created !@CD    by !@CB    Last updated !@UD    by !@UB
```

Figure 3. Maillist screen 1, The data entry screen.

then put field 4 (last name); 6) finally, put this concatenated string into dummy field "NM," an alphanumeric ("*") field which can be up to 39 characters long.

The operator that strings together text elements, is "⟨," which you'll recognize as the "push-left" indicator from Profile Plus and Profile III Plus label formats. The " = " sign does exactly what it does on Profile Plus and Profile III Plus math tables; it places the result of the operation on its right into the field specified on its left.

DUMMIES ARE SMART

A dummy, or "temporary," field can be up to 255 bytes long. Being temporary, it doesn't take up precious disk space for storage, but is reconstructed in a micro-second when each record is accessed. I could have placed the concatenated name fields into another "real" field, but then I would have traded a slight savings of time for an enormous waste of space.

When you create a dummy field, as I did in the first processing element, you assign it a name consisting of one or two letters. Letter-number combinations, such as "A1," are reserved for associated fields (see last month's column). You can assign a length and type to the dummy field, but this is not necessary — the program will assign it a length equal to the combined lengths of the fields it incorporates. There are three ways to truncate the length of a dummy field in printed output: 1) by specifying a shorter length when you define the dummy field; 2) by placing a backslash on the format (type ⟨CTRL 9⟩ or ⟨CTRL /⟩, in which case the field terminates at the backslash; and 3) by placing text or another field after it on the format, in which case the field terminates one space before the next printed character. Text is left-justified in dummy fields, but numeric fields are right-justified, so a little extra care is required. Use the backslash to assure that numbers align properly. If you truncate too severely, you might get an overflow message; however, the program still does math correctly as it "knows" the values involved. Dummy fields are placed on screens and in formats in the same way "real" fields are, with one of the field indicators ("*" or "!").

```
Jan  8 19:09 1984    File Name: maillist                      Page 1
                     Processing: labels

  1 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If: 8 EQ "H" OR 8 EQ "h"
      Then: GOTO HOME
  2 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
        If:
      Then: LT(37,*)=TD;LO(35,*)=6;LA(25,*)=14;LC(33,*)=CZ;END
  3 --------   -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
HOME    If:
      Then: LT="";LO="";LA(25,*)=19;LC(33,*)=HZ;END
```

Figure 7. The output processing table for labels.

STRIPTEASE

"〈" isn't the only kind of operator that links fields. "{" goes all the way, meaning it strips away all blanks, including the space between a field and its neighbor. I use this in automatic processing elements to add commas. Look at elements 5 and 6 in figure 2. For the office address in element 5 and the home address in element 6, I concatenate the city, state code and zip code fields so that:

New York NY 10013

becomes:

New York, NY 10013

The statement using the "{" operator strips all trailing blanks from the city field and adds a comma. Literals (characters, numbers and punctuation that you want to appear) must be enclosed in quotes. I use "〈" before state and zip because I want a space between them. In element 7, the "{" statement adds the comma to the salutation so that the person entering data doesn't have to bother with it.

CONDITIONS

Processing elements 1 and 2 are related to each other, as are elements 3 and 4, in that they specify complementary solutions to similar formatting problems.

If field 11, MS./Mr., is empty, the formula in element 1, "11〈5〈4," leaves a leading blank. To avoid this problem, I use condition lines to test for the presence or absence of an entry. Element 1 tests for the Ms./Mr.: 11 NE "", or field 11 is not equal to "" or null—it does contain data. In this case, it provides "11〈5〈4." If the record fails the test in element 1, processing moves to element 2, and the action in element 2 is executed: it provides "5〈4."

Paired elements 3 and 4 are a little more complicated. Field 12 is for the person's title (if any), and field 13 is for his or her department (if any).

The "if any" is the problem. Under normal circumstances, if both fields contain data, the usual concatenation process will produce:

Director, Purchasing

which is fine. But if only one or the other field contains data, the result will be either:

Director,

or:

Purchasing

both of which are not fine. The processing tables take literals literally, so to speak, and print a comma no matter what. And so I use the connective AND and OR on the condition lines to watch for abnormal circumstances so they can be dealt with appropriately.

INPUT PROCESSING

Figure 5 shows the input processing table for maillist. I set up the data base to record both the home and office addresses of each person. But I only want one mailing label to be generated when the time comes. I make my choice by entering either "H" or "O" in field 8. When I press 〈ESC〉〈ESC〉 to record or to update an entry, the program

```
Jan  8 21:35 1984    File Name: maillist                    Page 1
                     Processing: Input

 1 --------  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
       If: 8 EQ "H"
     Then: 3=22
 2 --------  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
       If: 8 NE "H"
     Then: 3=17
 3 --------  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
       If: @SN EQ "1"
     Then: INPUT XX(1,*) "Type <Y> if you need Code Entry Screen >"
 4 --------  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
       If: @SN EQ "1" AND (XX EQ "Y" OR XX EQ "y")
     Then: SCREEN 2
```

Figure 5. The input processing table.

puts either the home zip code (field 22) or the office zip code (field 17) into the mailing zip code (field 3). Notice that I use a "real" field here instead of a dummy field because Profile can only index and sort on real fields.

I MADE A MISTAKE

I want to tell you about a mistake I made in setting up these maillist processing tables because it shows the difference between automatic and input processing. (Books and manuals have their uses, but nothing beats making your own mistakes when you are trying to learn something new.) I originally put input processing elements 1 and 2 into the automatic processing table. I reasoned, logically enough, that I wanted to place the appropriate zip code into the mailing zip field "automatically." When I went back to test some sample entries, it seemed to work fine. But when I did my first real run of mailing labels of about 200 records, they did not come out in zip code order.

The reason for this turned out to be quite simple: field 3 was still blank for almost every record! Here's why: Automatic processing operates, as I said above, before a record is displayed on the screen. In adding records, all fields are, of course, blank until you fill them in for the first time. The next time you access these records, the automatic table will be able to process the information—in this case, move a home or office address into the mailing address field—but not until then. I had, however, generated my mailing list before reaccessing, and therefore reprocessing, these records. The sample records had worked only because I had accessed them all at least once during the testing process.

TRAFFIC CONTROL

Input processing elements 3 and 4 illustrate another wonderful capability of Profile 16, that of directing the course of data input.

As faithful readers of this column know, I love to use codes. But they become inefficient if the other people who have to deal with them have difficulty entering or deciphering them. Figure 4 shows a second data entry screen that explains the codes used in maillist. (The hardcopy prints dots where I use Profile's graphics characters to draw lines.)

I wanted to give the person entering data the option of switching from data entry screen 1 to code entry screen 2 without leaving add/update mode. This is a two-step operation.

"@SN" means screen number, and is one of a list of useful "system" fields maintained automatically by Profile 16. They are all indicated by a two characters preceded by "@". The "1" must be in quotes to distinguish it from field 1. Since the procedure I am describing makes sense only if the computer clerk is using screen 1, I test for this using "INPUT" and

"SCREEN," two processing table commands. The first allows you to display an instruction or question for the user (the message" enclosed in quotes) at the bottom of the display screen and to have the user's "answer" placed into a real or dummy field. In input processing element 3, I direct the answer to a one-character alphanumeric dummy field I call "XX."

Processing element 4 then tests field XX for the presence of a "Y" or "y." If the test passes, the action line sends the program to screen 2 without leaving update mode. This is very convenient. You can also call one of an additional 26 screen formats (A-Z) that are not otherwise accessible from Inquire, Update, Add.

You'll find many uses for INPUT. You can enter the current price of a fluctuating commodity or any other variable and then have Profile recalculate whatever values depend upon it. The user manual has many other suggestions.

OUTPUT WITH STYLE

Figure 6 shows the format screen for my maillist mailing labels. Both the format and the output processing table that works with it (figure 7) are called "labels." I use Radio Shack's two-column-wide, self-adhesive labels, which are 3.8" × 15/16" on a 9 1/2" wide carrier. Here's a handy tip for all Profile users. These labels have a gutter between them. If you tell the program that each label is 40 columns wide on an 80-column "page," the right-hand labels will print too far to the left. The trick is to specify 43-column labels on an 86-column carrier, and to make sure that no one address line is longer than 38 characters. These labels also have a one-line gutter above and below, so specify a six-line format but use only five lines. Adjust these numbers accordingly if you are using 12-pitch instead of 10-pitch. Proportional spacing is not supported for two-wide labels because the left margin of the right-hand labels will be uneven. Use the new one-wide labels instead.

```
Jan  8 19:17 1984  File Name: maillist          Format Name: labels          Page 1

        10        20        30        40
....:....|....:....|....:....|....:....|...
..............................D.A.T.A...L.I.N.E.S..............................
!NM
!LT
!LO
!LA
!LC

.............................E.N.D...O.F...F.O.R.M.............................
```

Figure 6. The labels format screen.

Profile 16 output formats allow you to send printer codes to your printer. The program comes supplied with all the codes for the DMP-2100, but can easily be configured for any printer. I use the correspondence quality 10-pitch font on my DMP-2100 by placing printer code 10 at the beginning of the first line. The labels look hand-typed.

QUIRKS

These printer codes have a few quirks. You can reset the font for standard-10 at the end of your label run by placing the proper printer code at the end of the last label line. This works e for one-wide labels. On two- or three-wide labels, however, the last line of the right hand labels will then print in the wrong font. This is because the program prints all two (or three) first lines, then all the second lines, etc. You'll have to reset the font at the end of the leftmost label's line, leaving the remaining last lines in the lurch. The solution to this minor problem is to set the correspondence quality font at the beginning of the last line (even though it has been set already) and to reset the standard font at the end of the last line.

(A personal note: I happen to enjoy discovering these pesky little quirks. The one just described is a perfectly logical consequence of the elegant and complex inner workings of Profile. The program did exactly what I told it to do. Not being as logical as the program, I thought I was telling it to do something slightly different. Live and learn; that's how you get smarter.)

YOUR PLACE OR MINE?

Getting back to my label format (figure 6), you'll notice it consists entirely of dummy fields. "NM" is the name field defined in automatic processing. There is no need to define it again, as procedures defined in automatic processing are used whenever records are used. The last four lines ("LT" for label title, "LO" for label organization, "LA" for label address and "LC" for label city/state/zip) are defined on the labels processing table (figure 7).

This table determines which of two addresses in each record to print on the mailing label, and then solves the following problem: If mail is directed to the person's office, the address is likely to be four or five lines long. If mail is sent to his or her home, the address is likely to be only three lines long, with title and organization left blank. Each record contains both addresses. The necessary branching happens in processing element 1. The condition line tests field 8 for an "H." If the test passes, the GOTO statement in the action line directs the flow of processing to element 3, which is labeled "HOME." Element 3 sets the second and third address lines to null, puts field 19 into the fourth (address) line and the formatted city/state/zip line from the automatic processing table into the last line. Notice that numerous processing instructions can be placed on the same action line, but they must be separated by semi-colons.

If the test in element 1 fails—that is, if mail is to be sent to the office—the processing drops down to element 2 and fills in each address line as appropriate. The "END" command stops processing and prevents it from continuing on to element 3. Since I instruct Request Output to remove unwanted blank lines (see "Requesting Output," Step 8, in the manual), the printed lines are pushed up to close the gap. The elegant results can be seen in figure 8.

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx        xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx                  xxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Jane Smith                                Mr. Ivan Sygoda
100 Main Street                           Author, Profile Column
Yourtown, NY 10000                        small Computer Company
                                          230 West 41st Street
                                          New York, NY 10036
```

Figure 8. Maillist mailing labels.

PROFILE Editor's Note: This is Mr. Sygoda's seventeenth article in a series of 'how-to' Profile articles. We hope that you enjoy this feature, and we look forward to your comments and questions on Profile.

Pentacle is a New York City-based non-profit service organization specializing in administrative services for performing art groups.

Calendar/Loan Calculator

Jay A. Jolicoeur
Switchback
Computer Programming Services
P.O. Box 427
Holden, MA 01501

Enclosed is a calendar program I first wrote while in college. The first Radio Shack version was for the Model III but I rewrote it for use on the Model II.

The comments in the program are self explanatory and the program will print a calendar accurately for any year after 1752. The only changes needed for use on a Model III are in line 70, the formatting of P$ and N$, and the deletion of line 345.

(Editor's Note: Mr. Jolicoeur also enclosed several other programs for consideration. We have chosen to publish his "Loan Calculator" program in addition to the "Calendar" program.)

Calendar

```
1Ø '***********************************************
2Ø ' CALENDAR PROGRAM IN BASIC FOR TRS-8Ø MODEL II
3Ø '           BY JAY A. JOLICOEUR 1981
4Ø '***********************************************
5Ø DIM M$(12),A$(42),B$(42),L$(42),N$(14)
6Ø DIM A(42),C(4),M(12)
7Ø C(1)=4
     : C(2)=2
     : C(3)=Ø
     : C(4)=6
     : P$="  \\"
     : N$="\       \ ####"
8Ø FOR K=1 TO 12
     : READ M(K)
     : READ M$(K)
     : NEXT K
9Ø DATA1,"JANUARY",4,"FEBRUARY",4,"MARCH",Ø,"APRIL"
1ØØ DATA2,"MAY",5,"JUNE",Ø,"JULY",3,"AUGUST",6,
    "SEPTEMBER"
11Ø DATA1,"OCTOBER",4,"NOVEMBER",6,"DECEMBER"
12Ø FOR K=1 TO 31
     : READ L$(K)
     : NEXT K
13Ø DATA" 1"," 2"," 3"," 4"," 5"," 6"," 7"," 8",
    " 9","1Ø"
14Ø DATA"11","12","13","14","15","16","17","18",
    "19","2Ø"
15Ø DATA"21","22","23","24","25","26","27",
    "28","29","3Ø","31"
16Ø '***********************************************
17Ø 'THIS PROGRAM WAS DERIVED FROM A FORMULA TO FIND
18Ø 'THE DAY OF THE WEEK OF ANY DATE:
19Ø '(1) TAKE THE LAST TWO DIGITS OF THE YEAR
2ØØ '(2) ADD 1/4 OF THIS NUMBER, NEGLECTING ANY
    REMAINDER
21Ø '(3) ADD THE DATE IN THE MONTH
22Ø '(4) ADD ACCORDING TO THE MONTH:
23Ø 'JAN 1(Ø IN LEAP YEAR) - FEB 4(3 IN LEAP YEAR)
24Ø '  MAR 4 - APR Ø - MAY 2 - JUN 5 - JUL Ø
25Ø '  AUG 3 - SEP 6 - OCT 1 - NOV 4 - DEC 6
26Ø '(5) ADD 4 FOR THE 18TH CENT;
    ADD 2 FOR THE 19TH CENTURY
27Ø 'ADD Ø FOR THE 2ØTH CENT;ADD 6 FOR THE 21ST CENT
28Ø '      REPEAT FOR SUBSEQUENT YEARS
29Ø '(6) DIVIDE THE TOTAL BY 7. THE REMAINDER GIVES
3ØØ '  THE DAY OF THE WEEK AS FOLLOWS:
31Ø '1 SUN, 2 MON, 3 TUE, 4 WED, 5 THU, 6 FRI, Ø SAT
32Ø 'THIS FORMULA IS CORRECT FOR ANY DATE AFTER
33Ø '  14 SEPTEMBER 1752
34Ø '***********************************************
345 CLS
     : SYSTEM "FORMS (L=66)"
35Ø CLS
     : PRINT "ADJUST PAPER TO TOP OF PAGE"
     : PRINT
36Ø LINE INPUT "YEAR TO BE PRINTED
     :  ";Y$
     : Y=VAL(Y$)
     : IF Y=-1 THEN END
37Ø '***********************************************
38Ø 'DETERMINE WHAT YEAR IT IS AND
      IF IT IS A LEAP YEAR *
39Ø '***********************************************
4ØØ I=Y-FIX(Y/4)*4
     : J=Y-FIX(Y/1ØØ)*1ØØ
41Ø IF I<>Ø THEN 43Ø
     : IF J=Ø THEN 43Ø
42Ø M(1)=Ø
     : M(2)=3
43Ø Y1=Y
     : Y=Y-FIX(Y/1ØØ)*1ØØ
44Ø K=(Y1-Y)/1ØØ-16
     : K=K-FIX(K/4)*4
     : C1=C(K)
45Ø FOR I=1 TO 11 STEP 2
46Ø '****************
47Ø '* ODD # MONTHS *
48Ø '****************
49Ø IF I=1 OR I=3 OR I=5 OR I=7 THEN N=31
5ØØ IF I=9 OR I=11 THEN N=3Ø
51Ø D=Y+.25*Y
     : D=FIX(D+1)
     : D=D+M(I)
     : D=D+C1
52Ø D=D-FIX(D/7)*7
     : IF D=Ø THEN D=7
53Ø FOR J=1 TO 42
     : A(J)=Ø
     : NEXT J
54Ø ON D GOSUB 81Ø,82Ø,83Ø,84Ø,85Ø,86Ø,87Ø
55Ø FOR J=1 TO 42
     : A$(J)=L$(A(J))
     : NEXT J
56Ø '*****************
57Ø '* EVEN # MONTHS *
58Ø '*****************
59Ø B=I+1
6ØØ IF B=2 THEN N=28
     : IF M(2)=3 THEN N=29
61Ø IF B=4 OR B=6 THEN N=3Ø
62Ø IF B=8 OR B=1Ø OR B=12 THEN N=31
63Ø D=Y+.25*Y
     : D=FIX(D+1)
     : D=D+M(B)
     : D=D+C1
64Ø D=D-FIX(D/7)*7
     : IF D=Ø THEN D=7
65Ø FOR J=1 TO 42
     : A(J)=Ø
     : NEXT J
66Ø ON D GOSUB 81Ø,82Ø,83Ø,84Ø,85Ø,86Ø,87Ø
67Ø FOR J=1 TO 42
     : B$(J)=L$(A(J))
     : NEXT J
68Ø '*****************************
69Ø '* PRINT MONTHS - 2 AT A TIME *
7ØØ '*****************************
71Ø LPRINT TAB(7);
     : LPRINT USING N$;M$(I);Y1;
     : LPRINT TAB(5Ø);
     : LPRINT USING N$;M$(B);Y1
```

```
72Ø LPRINT TAB(7)"--------------";
     : LPRINT TAB(5Ø)"--------------"
73Ø LPRINT" SUN MON TUE WED THU FRI SAT";
     : LPRINT TAB(43)"SUN MON TUE WED THU FRI SAT"
74Ø LPRINT" ---------------------------";
     : LPRINT TAB(43)"---------------------------"
75Ø E=1
     : F=7
     : GOSUB 79Ø
     : E=8
     : F=14
     : GOSUB 79Ø
76Ø E=15
     : F=21
     : GOSUB 79Ø
     : E=22
     : F=28
     : GOSUB 79Ø
77Ø E=29
     : F=35
     : GOSUB 79Ø
     : E=36
     : F=42
     : GOSUB 79Ø
78Ø NEXT I
     : LPRINT CHR$(12)
     : GOTO 35Ø
79Ø FOR J=E TO F
     : LPRINT USING P$;A$(J);
     : NEXT J
     : LPRINT"                ";
     : FOR J=E TO F
     : LPRINT USING P$;B$(J);
     : NEXT J
     : LPRINT
8ØØ RETURN
81Ø FOR J=1 TO N
     : A(J)=J
     : NEXT J
     : RETURN
82Ø FOR J=2 TO N+1
     : A(J)=J-1
     : NEXT J
     : RETURN
83Ø FOR J=3 TO N+2
     : A(J)=J-2
     : NEXT J
     : RETURN
84Ø FOR J=4 TO N+3
     : A(J)=J-3
     : NEXT J
     : RETURN
85Ø FOR J=5 TO N+4
     : A(J)=J-4
     : NEXT J
     : RETURN
86Ø FOR J=6 TO N+5
     : A(J)=J-5
     : NEXT J
     : RETURN
87Ø FOR J=7 TO N+6
     : A(J)=J-6
     : NEXT J
     : RETURN
```

Loan

```
1Ø '****************************************
2Ø '* LOAN CALCULATOR FOR TRS-8Ø MODEL II *
3Ø '* Revisions by Jay A. Jolicoeur 1982  *
4Ø '****************************************
5Ø CLEAR 1ØØ
     : CLS
     : S$=CHR$(26)
     : S1$=CHR$(25)
     : DEFINT J,L,N
     : DEFDBL A-F,M,P-Z
     : DEFFNZ$(Z1$)=S$+" "+Z1$+" "+S1$+" "
6Ø CLS
     : PRINT TAB(8)S$ "LOAN CALCULATOR "S1$
     : PRINT
     : LINE INPUT "LOAN AMOUNT......$";A$
     : A=VAL(A$)
     : GOSUB 35Ø
     : IF A=Ø THEN 6Ø
7Ø LINE INPUT "INTEREST RATE....%";R$
     : R=VAL(R$)
8Ø LINE INPUT "LENGTH OF LOAN... YEARS> ";Y$
     : LINE INPUT"                 MONTHS> ";M1$
     : Y=VAL(Y$)
     : M1=VAL(M1$)
     : N=Y*12+M1
     : R=ABS(R)
     : M=R/12ØØ
     : GOSUB 34Ø
9Ø W=(1+M)^N
     : P=(A*M*W)/(W-1)
     : P=INT(P*1ØØ+.99)
     : P=P/1ØØ
     : PRINT USING "MONTHLY PAYMENT IS $$#,###.##";P
1ØØ FP=P
11Ø PRINT@(15,Ø),FNZ$( "1") "SHOW MONTHLY
    ANALYSIS ";
    : PRINT@(16,Ø),FNZ$("2") "OVERRIDE MONTHLY
    PAYMENT ";
    : PRINT@(17,Ø),FNZ$("3") "START OVER";
    : PRINT@(18,Ø),FNZ$("4") "TOTALS";
    : PRINT@(19,Ø),FNZ$("5") "PRINTOUT OF
    MONTHLY ANALYSIS";
115 PRINT@(2Ø,Ø),FNZ$("F1") "END";
    : PRINT@(21,Ø),"";
12Ø C$=INKEY$
    : IF C$="" THEN 12Ø ELSE IF C$=CHR$(1)
    THEN END ELSE PRINT C$
    : C=VAL(C$)
13Ø ON C GOTO 15Ø,14Ø,5Ø,17Ø,165
    : GOTO 11Ø
14Ø PRINT@(1Ø,Ø),"";
    : LINE INPUT "MONTHLY PAYMENT..$";P$
    : P=VAL(P$)
    : GOTO 1ØØ
15Ø GOSUB 5ØØ
165 IF C=5 THEN GOSUB 4ØØ
17Ø B=A*1ØØ
    : TT=Ø
    : TP=Ø
    : L=Ø
    : P=P*1ØØ
    : R$=""
18Ø FOR J=1 TO N
    : T=M*B
    : T=INT(T+.5)
19Ø IF J=N THEN P=B+T
2ØØ TP=TP+P
    : B=B-P+T
    : TT=TT+T
21Ø IF B<Ø THEN GOSUB 36Ø
22Ø IF R$="T" THEN 29Ø ELSE IF C=4 THEN 29Ø
23Ø PB=B/1ØØ
    : PT=T/1ØØ
    : T2=TT/1ØØ
235 IF C=5 THEN GOSUB 6ØØ
    : NEXT
    : GOTO 295
24Ø PRINT TAB(8) USING "###";J;
    : PRINT TAB(14) USING "#,###.##  ";P/1ØØ;
    : PRINT USING "####,###.##-";PB;PT;T2
25Ø IF B=Ø THEN J=N
    : GOTO 26Ø ELSE L=L+1
    : IF L<12 THEN 29Ø
26Ø PRINT@(18,Ø),FNZ$("T") "TOTALS "FNZ$("F2")
    "HARD COPY "FNZ$("ENTER") "TO CONTINUE";
```

```
27Ø R$=INKEY$
     : IF R$="" THEN 27Ø ELSE IF R$=CHR$(2)
     THEN PRINT@(18,Ø),CHR$(23);
     : SYSTEM "SCREEN"
     : LPRINT CHR$(12)
     : GOTO 26Ø
28Ø L=Ø
     : GOSUB 5ØØ
29Ø NEXT
295 IF C=5 THEN LPRINT
     : LPRINT TAB(1Ø) USING "LAST PAYMENT
     =$$###,###.##";P/1ØØ
     : LPRINT TAB(8) USING "TOTAL PAYMENTS
     =$$###,###.##";TP/1ØØ
     : LPRINT TAB(7) USING "MONTHLY PAYMENT
     =$$###,###.##";FP
     : LPRINT CHR$(12)
     : L=Ø
     : GOSUB 34Ø
     : GOTO 11Ø
3ØØ PRINT@(4,Ø),"";
     : PRINT TAB(1Ø) USING "LAST PAYMENT
     =$$###,###.##";P/1ØØ
     : PRINT TAB(8) USING "TOTAL PAYMENTS
     =$$###,###.##";TP/1ØØ
     : PRINT TAB(7) USING "MONTHLY PAYMENT
     =$$###,###.##";FP
31Ø PRINT@(18,Ø),FNZ$("F2") "HARD COPY "FNZ$("ENTER")
     "TO CONTINUE ";
32Ø R$=INKEY$
     : IF R$="" THEN 32Ø ELSE IF R$=CHR$(2) THEN
     FOR J5=13 TO 23
     : PRINT@(J5,Ø),CHR$(23);
     : NEXT J5
     : SYSTEM "SCREEN"
     : LPRINT CHR$(12)
     : GOTO 31Ø
33Ø PRINT@(18,Ø),CHR$(23);
     : P=FP
     : GOTO 1ØØ
34Ø CLS
     : PRINT USING "$$###,###.## FOR ## YEARS
     AND ## MONTHS AT ##.##% INTEREST";A;Y;M1;R
     : RETURN
35Ø A=INT(ABS(A))
     : IF A<1ØØØØØØØ THEN RETURN ELSE PRINT S$
     " TOO LARGE "S1$
     : A=Ø
     : RETURN
36Ø P=P+B
     : TP=TP+B
     : B=Ø
     : RETURN
4ØØ LPRINT USING "$$###,###.## FOR ## YEARS AND
     ## MONTHS AT ##.##% INTEREST";A;Y;M1;R
     : LPRINT TAB(44) "INTEREST PAID"
     : LPRINT TAB(7) "MONTH";TAB(15) "PAYMENT";
     TAB(28) "BALANCE";TAB(42) "MONTH";TAB(56) "YTD"
     : LPRINT TAB(7) STRING$(52, "-")
     : RETURN
5ØØ GOSUB 34Ø
     : PRINT TAB(44) "INTEREST PAID"
     : PRINT TAB(7) "MONTH";TAB(15) "PAYMENT";
     TAB(28) "BALANCE";TAB(42) "MONTH";TAB(56) "YTD"
     : PRINT TAB(7) STRING$(52,"-")
     : RETURN
6ØØ LPRINT TAB(8)USING "###";J;
     : LPRINT TAB(14) USING "#,###.## ";P/1ØØ;
     : LPRINT USING "####,###.##-";PB;PT;T2
     : L=L+1
     : IF L>5Ø THEN L=Ø
     : LPRINT CHR$(12)
     : GOSUB 4ØØ
     : RETURN ELSE RETURN
```

Disk ID

Alfred Pacheco
2203 Lincoln Pl. S.E.
Renton, WA 98055

Occasionally, I have felt the need of a program which would automatically check the disk drives to insure that each drive contains the proper disk before reading or writing any information. Although BASIC does not contain a command which allows you to do this conveniently, the Model II does contain a Supervisor Call which allows you to obtain the Disk ID. I am enclosing an assembly language program which will allow this Supervisor Call (Function Call 15) to be used from a BASIC program. I have also included a sample BASIC program which uses the subroutine.

Listing 1 shows the assembly language program with comments showing how it handles the interface between the BASIC program and Function Call 15. No error checking or handling routine has been included. The user must take proper precautions to insure that the proper parameters are passed to the program.

Listing 1

```
1ØØ    INC   DE        ;SET REG DE TO POINT TO
                        LOCATION OF
2ØØ    EX    DE,HL     ;ADDRESS OF STRING STORAGE AREA
                        (SSA)
3ØØ    LD    E,(HL)    ;LOAD ADDRESS OF SSA INTO REG HL
4ØØ    INC   HL        ;
5ØØ    LD    D,(HL)    ;
6ØØ    EX    DE,HL     ;
7ØØ    LD    A,(HL)    ;GET FIRST STRING CHAR INTO REG A
8ØØ    SUB   3ØH       ;CONVERT STRING NUMBER INTO
                        BINARY
9ØØ    LD    B,A       ;LOAD DRIVE NUMBER INTO B REG
1ØØØ   LD    A,15      ;GET DISKID
11ØØ   RST   8         ;
12ØØ   RET             ;RETURN TO BASIC PROGRAM
13ØØ   END
```

Listing 2 shows how DEBUG can be used to enter the program into your system. To do this, follow these steps:

1. At TRSDOS READY, type: DEBUG ON. After the response "DEBUG ON", enter: DEBUG.
2. In the DEBUG mode, enter: M. The display will show "M A=". Type: EF00.
3. Press the [F1] key.
4. Enter the hex codes as follows: 13 EB 5E 23 56 EB 7E D6 30 47 3E 0F CF C9 (See Listing 2).
5. Press the [F2] key.
6. Press the [S] key.
7. You will be back at TRSDOS READY. Type: DUMP DISKID [START=EF00 END=EF0D]
8. Type: DEBUG OFF.

Listing 2

```
EF00    13 EB 5E 23 56 EB 7E D6   30 47 3E 0F CF C9 00 00   ..^#V...0G>.....
EF10    00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
EF20    00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
EF30    00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
EF40    00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
EF50    00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
EF60    00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
EF70    00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
   PC    SP  SZHPNC  AF   BC   DE   HL   IX   IY   AF'  BC'  DE'  HL'
 2800  21FE  000000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
? P
DEBUG is now ON
TRSDOS READY
DEBUG
```

Listing 3 is the sample BASIC program using the subroutine. The following program description should help in interfacing it to your own particular program.

Line 10: Some form or variation needed to protect the machine program.
Line 20: Loads machine program into memory.
Line 30: Tells BASIC where routine is located.
Line 40: Clears space for Disk ID (eight spaces).
Line 50: Sets C$ so that it contains drive number (0, 1, 2, or 3). The drive number must be a string character (one can use C$ = "0" for example).
Line 60: Puts drive number into string A$.
Line 70: Call routine.
Line 80: A$ contains DISKID. Checks for right ID by - IF A$ = "TRSDOS" THEN GOTO . . . Remember A$ has eight characters.

Listing 3

```
10 CLEAR 100,61183
20 SYSTEM "LOAD DISKID"
30 DEFUSR0=&HEF00
40 A$="        "
50 INPUT C$
60 MID$(A$,1,1)=C$
70 A$=USR0(A$)
80 PRINT A$
```

ReName for Profile II/ Profile Plus

Calvin Roberts
P.O. Box 22413
San Diego, CA 92122

Enclosed is a listing of a program I have developed for users of Profile II and Profile Plus. *(Editor's Note: We tested the Program on our Model II Profile Plus.)* I also used a DO file, "NAME", which can be incorporated into the User Menu or called from TRSDOS and is structured to allow the user to modify or change the program to enhance or tailor it to fit personal or business needs. (See Listing 1)

Listing 1

BUILD NAME (ENTER)
BASIC RENAME/BAS (ENTER)
(BREAK)

I have found that when I create a filing system I often use the same formats or slight variations of a previous one. I usually use this program to RENAME filing, mailing, report, etc., systems that were created and stored in a library of formats. These formats can be MOVED, using the wild card filespec on to a working disk. The new files can be modified to fit the application called for at the time, and then RENAMED, using this program, thus avoiding the need to RENAME each file separately.

Also, when doing housekeeping and making changes or modifications to a data base, the filename can be shortened to one or two characters to keep from having to input a long filename each time. After the changes have all been made, the filename can then be restored to the original filename.

```
1000 ' October 24th, 1982
1100 ' Calvin Roberts
1200 ' P.O. Box 22413
1300 ' San Diego, CA 92122
1400 '
1500 ' This program was created for the
1600 ' PROFILE II and the PROFILE PLUS
1700 ' users to RENAME files.
1800 '
1900 ' It also allows the user to create a
2000 ' library of commonly used formats,
2100 ' stored on a disk, which can be moved
2200 ' using the MOVE filename*/*:d TO :d
2300 ' (see operators manual for use of
2400 '  the WILD CARD filespec) from one
2500 ' drive to another, then this program
2600 ' can RENAME all of the files to the
2700 ' working PROFILE file name.
2800 '
2900 ' The segments, screen, report, label,
3000 ' etc., formats can then be tailored
3100 ' to fit the particular application
3150 ' involved.
3200 '
3300 ' Under "Data Statements" are all of
3400 ' the extents used by PROFILE PLUS
3500 ' (see PROFILE II or PROFILE PLUS
3600 ' manuals).  Extents never used in
3650 ' the user's files or in the user's
3700 ' library can be deleted from the
3750 ' Data Statements.
3800 '
3900 ' If the user deletes or adds extents
4000 ' to the Data Statements be sure that
4100 ' a 999 is the last data record in the
4200 ' group as this is the EOF (end of file
4250 ' marker).
4300 '
4350 ' If you make changes or enhancements
4351 ' please send me a copy and a brief
4352 ' description of the changes and what
4353 ' each does, at the above address.
4354 ' This information may helpful to
4355 ' to other users in other applications.
4356 '
4400 '
4500 '    LIST OF STRING VARIABLES
4600 '
4700 ' A$  - Variable for holding inputted old
     PROFILE file name
4800 ' Also used for holding old file name padded to
     eight characters
4900 ' B$  - Variable for holding inputted new
     PROFILE file name
5000 ' Also used for holding new file name padded to
     eight characters
5100 ' C$  - Temporary variable for reading the
     extent data
5200 ' D$  - Variable for holding the completed old
     file name
5300 ' E$  - Variable for holding the completed new
     file name
5400 ' F$  - Variable for holding the file name that
     has been renamed
5500 ' G$  - Variable for holding the file name that
     has not been found
5600 ' X$  - Temporary variable for holding file
     names when
5700 ' padding to eight characters
5800 ' X1$ - Temporary variable for holding the
     correct number of 0's
5900 ' for padding to eight characters
6000 ' Z$  - Temporary variable for holding file
     names when
6100 ' padding to eight characters
6200 '
```

```
6300 '
6400 '
6500 '    LIST OF NUMERICAL VARIABLES
6600 '
6700 ' A  - Counter variable for number of files
       renamed
6800 ' B  - Counter variable for number of files not
       renamed (not found)
6900 ' C  - Indicator variable
7000 ' J  - Counter variable
7100 ' K  - Counter variable
7200 ' L  - Variable for determining max number of
       files
7300 ' V  - Counter variable
7400 ' X  - Temporary variable for holding length of
       PROFILE file name
7500 ' X1 - Temporary variable for holding number of
       Ø's to pad file with
7600 ' V  - Menu selection variable
7700 '
7800 '
7900 '
8000 ' Program Start
8100 '
8200 CLEAR 4000
8300 ON ERROR GOTO 23000
8400 DIM D$(50)
8500 DIM E$(50)
8600 DIM C$(50)
8700 DIM F$(50)
8800 DIM G$(50)
8900 '
9000 '
9100 '
9200 ' Menu
9300 '
9400 CLS
     : A=Ø
     : B=Ø
     : PRINT TAB(25);"PROFILE FILE MANIPULATION
     MENU"
9500 A$=" "
     : B$=A$
     : FOR J=Ø TO 5Ø
     : C$(J)=A$
     : D$(J)=A$
     : E$(J)=A$
9600 F$(J)=A$
     : G$(J)=A$
     : NEXT
     : J=Ø
9700 PRINT
     : PRINT
9800 PRINT TAB(3Ø);"1 - Single File Rename"
     : PRINT
9900 PRINT TAB(3Ø);"2 - Total File Rename"
     : PRINT
10000 PRINT TAB(3Ø);"3 - Profile II Menu"
     : PRINT
10100 PRINT TAB(3Ø);"4 - Exit to TRSDOS"
     : PRINT
10200 PRINT
     : PRINT
     : PRINT "Enter Selection :";
10300 INPUT Y
     : ON Y GOTO 11000,12100,12900,13400
10400 GOTO 9400
10500 '
10600 '
10700 '
10800 ' Single File Rename
10900 '
11000 GOSUB 14000
11100 PRINT "Extent :";
11200 LINE INPUT C$
     : PRINT
11300 GOSUB 14800
11400 D$(Ø)=A$+"/"+C$
     : E$(Ø)=B$+"/"+C$
11500 L=Ø
     : GOSUB 20200
11600 PRINT
     : GOSUB 22400
     : GOTO 9400
11700 '
11800 '
11900 ' All File Rename
12000 '
12100 GOSUB 14000
     : GOSUB 14800
12200 GOSUB 17700
     : GOSUB 18900
12300 GOSUB 20200
     : GOSUB 21400
12400 GOSUB 22400
     : GOTO 9400
12500 '
12600 '
12700 ' Exit to PROFILE II Menu
12800 '
12900 SYSTEM "M"
13000 '
13100 '
13200 ' Exit to TRSDOS
13300 '
13400 SYSTEM
13500 '
13600 '
13700 '
13800 ' Subroutine to Input Old PROFILE File Name
13900 '
14000 CLS
     : PRINT "Old PROFILE File Name :";
14100 LINE INPUT A$
14200 PRINT
     : Z$=A$
     : GOSUB 23900
     : A$=Z$
     : RETURN
14300 '
14400 '
14500 '
14600 ' Subroutine to Input New PROFILE File Name
14700 '
14800 PRINT "New PROFILE File Name :";
14900 LINE INPUT B$
15000 PRINT
     : Z$=B$
     : GOSUB 23900
     : B$=Z$
15100 '
15200 ' Check to see if Old and New File Names
     Are the Same
15300 '
15400 IF A$<>B$ THEN GOTO 15700
15500 PRINT "Files Cannot Have The Same File
     Names !!"
15600 FOR V=1 TO 1000
     : NEXT
     : GOTO 9400
15700 PRINT
     : RETURN
15800 '
15900 '
16000 '
16100 ' DATA Statements
16200 '
16300 DATA MAP,KEY,DAT,DA2,DA3:'  File Segments
16400 DATA PMP,PM2,PM3,PM4,PM5:'  Screen Formats
16500 DATA PRT,PR2,PR3,PR4,PR5:'  Report Formats
16600 DATA LB1,LB2,LB3,LB4,LB5:'  Label Formats
16700 DATA SL1,SL2,SL3,SL4,SL5:' Selection Formats
```

```
16800 DATA SR1,SR2,SR3,SR4,SR5:' Selection Record
    Files
16900 DATA KX1,KX2,KX3:'         Expanded Files
17000 DATA IX1:'                 Index File
17100 DATA 999:'                 End of File Marker
17200 '
17300 '
17400 '
17500 ' Subroutine to Add Extents to Old File Names
17600 '
17700 READ C$(J)
17800 IF C$(J)="999" THEN 18300
17900 D$(J)=A$+"/"+C$(J)
18000 J=J+1
18100 GOTO 17700
18200 L=J-1
18300 RESTORE
     : RETURN
18400 '
18500 '
18600 '
18700 ' Subroutine to Add Extents to New File Names
18800 '
18900 J=0
19000 READ C$(J)
19100 IF C$(J)="999" THEN 19500
19200 E$(J)=B$+"/"+C$(J)
19300 J=J+1
19400 GOTO 19000
19500 L=J-1
19600 RESTORE
     : RETURN
19700 '
19800 '
19900 '
20000 ' Subroutine to RENAME Files
20100 '
20200 FOR K=0 TO L
20300 NAME D$(K) AS E$(K)
     : IF C=1 THEN 20600
20400 PRINT D$(K);" ------> ";E$(K)," Completed"
20500 A=A+1
     : F$(A)=D$(K)
20600 C=0
20700 NEXT
     : RETURN
20800 '
20900 '
21000 '
21100 ' Subroutine to Inform User of Number and
21200 '  Files RENAMED and Files NOT FOUND
21300 '
21400 CLS
     : PRINT "FILES RENAMED: ";A," ","FILES NOT
     FOUND: ";B
21500 PRINT
21600 FOR L=1 TO J STEP 2
21700 K=L+1
     : PRINT F$(L),F$(K)," ",G$(L),G$(K)
21800 NEXT
     : RETURN
21900 '
22000 '
22100 '
22200 ' Pause Subroutine
22300 '
22400 PRINT "Hit <ENTER> to Continue :";
     : INPUT V
     : RETURN
22500 '
22600 '
22700 '
22800 ' Error Subroutine for FILE NOT FOUND
22900 '
23000 IF ERR<>53 THEN 24600
23100 PRINT "File ";D$(K);" NOT on Disk"
23200 B=B+1
     : G$(B)=D$(K)
     : C=1
23300 RESUME NEXT
23400 '
23500 '
23600 '
23700 ' Subroutine to Pad PROFILE File Name to Eight
     Characters
23800 '
23900 X=LEN(Z$)
     : X1=8-X
     : X1$=STRING$(X1,"0")
24000 X$=Z$+X1$
     : Z$=X$
     : X=0
     : X1=0
     : RETURN
24100 '
24200 '
24300 '
24400 ' Error Subroutine for All Other Errors
24500 '
24600 PRINT "ERROR Number ";ERR;" Occurred at Line
     Number ";ERL
     : PRINT
24700 PRINT "Do you wish to resume "
     : INPUT Y$
24800 IF Y$<>"N" THEN 23300
24900 END
```

Computer Clubs

CAPATUG
Capital Area TRS-80 Users Group
340 Lewisberry Rd.
New Cumberland, PA 17070
(717) 469-0567

THE COLOR FORCE
Reid Baker
Dunedin Realty
503 South Paula Drive.
Dunedin, FL 33528
(813) 733-5095

SUPER USER
c/o Bill Clemo
M & M Technologies Corp
P.O. Box 237
Herndon, PA 17830
(717) 758-9260
(Model 16-Xenix Users Group)

68000 SOFTWARE USERS GROUP
c/o Carl Cagan
Data Management Systems
211 N. El Camino Real, Ste. 101C
Encinitas, CA 92024
(619) 942-0744

Forum Update: What's New in the Professional Circle

Editor's Note: The Consumer Information Service is one of the largest information and entertainment services available to owners of personal computers and computer terminals. With each issue of TRS-80 Microcomputer News, *various features of CompuServe will be discussed. The CompuServe Consumer Information Service is sold at Radio Shack stores nationwide and in Canada.*

Many of us think of Honolulu as that distant paradise where waves break gently on Waikiki's palm lined beaches and where warm sunshine and fragrant trade winds are an everyday occurrence.

"Most people who come here for vacation see only our island's natural beauty," says Robert Brown, a systems consultant, industry observer and editor of the Direct Connection® Newsletter which has recently been added to the CompuServe system.

"They don't realize that the capital of the 50th state is also becoming a very active regional business center," Brown remarked.

Brown, who has made his home in Hawaii for the last three years, enjoys producing the newsletter for fellow computer and communications professionals and adds that publishing in Honolulu just makes it that much more pleasant. Originally printed on paper and distributed via first class mail, the Direct Connection has evolved to a purely electronic publication with nationwide "Help Wanted" advertising, an industry news section that regularly "scoops" other conventional computer and communications publications and an editorial emphasis on the human aspect of the information revolution.

Consumer Information Service subscribers should type GO TDC from any prompt and Executive Information Service subscribers should type GO DCT from any prompt to read the Direct Connection.

FROM THE COMMUNICATIONS INDUSTRY FORUM

The Communications Industry Forum was created to provide the communications professional a means to exchange ideas, thoughts, information and concerns about the industry. The Forum deals with both the technical and aesthetic sides of electronic communications. This includes radio, television, telecommunications, law, promotion, advertising, journalism and other forms of communication.

System operators report conferences to date on AM stereo and electronic editing. More conferences are in the works, and the membership roster is more than 1,300 strong. Members include broadcast engineers, journalists, communications engineers, marketing and promotional people to name a few.

MEDSIG CONTINUES GROWTH

The MEDSIG is designed to provide a forum for AAMSI members, subscribers of CompuServe, other interested health care professionals and people in other related fields of study.

The MEDSIG consists of 50 professional specialty groups. Each group determines the activities in which they will participate, i.e., conferences, dialogue, software information discussions, etc.

Gordon Black, system operator for the MEDSIG, reports that the American Association for Medical Systems and Informatics (AAMSI) conference held in October featured a workshop where CompuServe demonstrated its VIDTEX software and discussed several other options for the future of "telematics" in the medical profession.

HIGHLIGHTS FROM THE ENVIRONMENTAL FORUM

Here are a few things the Environmental Forum would like to highlight.

The Environmental Forum serves as a national clearinghouse for the sharing of information and experiences concerning environmental issues. Whether you are a professional consultant, government official, equipment manufacturer, student or concerned individual, there is something here for you. Membership to the Forum is free and open to all interested users of the Consumer Information Service.

Regular Conferences: Informal roundtable discussions of various environmental topics will be a standard feature beginning at 9:30 p.m. EST on Tuesdays.

New Database: This addition discusses natural/biological and chemical pesticides for the home and garden. The database will coincide with the growing season, with intensions to highlight a different type of chemical or natural killer each week. The weekly highlight will depend on what type of pest is "popular" at the time.

The Creature Feature: The animal of the week feature is a short, weekly article on wildlife and endangered species. The Creature Feature is designed to promote public awareness.

AVIATION SAFETY INSTITUTE AND AVIATION SPECIAL INTEREST GROUP

New. Twice a month, every month. The "ASI Monitor" newsletter offers news highlights from the world of aviation safety. Monitors are available from 1980 to the present. Find out what has happened to the airlines throughout their 80-year history. This is reading for every pilot by John B. Galipault, president and founder of the Aviation Safety Institute.

New. Guest editorials from the field by those who live and work on the front lines. Here you can find out why "The

Emperor Wears No Clothes," "The Testimony Before the Levitas Subcommittee" and why "Horse Feathers Don't Fly."

New. On-line airport reviews which are updated as of Jan. 1, 1984. Where's the best airport to land for gas? Good service? Where can you get a good meal for the family, and what airfields should you avoid?

Find out important facts about airports east and west of the Mississippi, in the mountains, the Plains and the most northern airports of Canada. Feedback submitted by AVSIG members flying the country make this a source of information when planning a cross country trip for business or pleasure.

New. Why is there no radar for South Bend, Ind. pilots? Find out flying conditions in southwest lower Michigan and northern Indiana by reading about the missing link in the AVSIG X7 database.

Also, turn to the Aviation Safety Institute features for the latest news on service reports for single- and multi-engine aircraft, safety tips such as the threat of migratory birds, spiders in your space and singing strut syndrome, and human factors such as eyes, pilot fatigue, cockpit choreography or preparing for forced landings.

All this and more for '84. Access the Services For Professionals and find out all you can about all the programs for those who want and need to stay in front of their profession.

Questions and comments about the CompuServe Consumer Information Service can be sent to Richard A. Baker, Editorial Director, or Jacqueline A. Farthing, assistant editor, CompuServe Information Services, 5000 Arlington Centre Boulevard, P.O. Box 20212, Columbus, Ohio 43220, or through Feedback, User Information section.

Notes On Previous Issues

JANUARY 1984

Bugspray

Dean Hildebrandt
30 Lincoln Lane
Simsbury, CT 60670

This program was incorrectly identified as a Color Computer program. It is for Model I/III/4 systems.

(Editor's Note: Our sincere apologies to the author and the readers who wrote in regarding this error. We continually strive for improvement; however, we are as yet still human. Our thanks and gratitude to our divine readers who point out but forgive our errors.)

Create a Random File from BASIC

Donald P. Raridan
963 Johnfer Way
Sacramento, CA 95831

Whoops! I failed to test all of my error traps. As a result, I discovered that the drive number input trap does not work. The reason is immediately obvious. The following lines were affected and appear here in their corrected form.

(Editor's Note: Our thanks to Mr. Raridan for his followup note regarding the problem in his program. Unfortunately, the information reached us too late to make the corrections before going to print.)

```
40 CLEAR 5000
     : CLS
     : DEFINT A-Z
     : DIM BR(30),BF$(30),PF$(30)
     : ON ERROR GOTO 1330
160 PRINT
     : INPUT  "ON WHAT DRIVE IS FILE LOCATED OR
    TO BE CREATED:";DR$
170 IF L>0 THEN 250
180 IF F$="E" THEN 200
```

Also, the following line should be added to the program.

```
1330 IF ERR=56 THEN 1320 ELSE 1300
```

Time/Date Display

Gordon Osmundson
475 North St.
Oakland, CA 94609

Here are some changes for Yoichi Ishikawa's interesting Time/Date Display program that will change it to a digital clock and date display.

Change Line 10 to:

```
10 ARUN:WAIT 0
```

Add:

```
130 GOTO 30
```

Upgrading Profile III+ to LDOS Floppy

A warning should have accompanied this article. Radio Shack Model III drives are only certified to format 40 cylinders. We tested the process on our system and it worked; however, we cannot say whether or not it will work on your system.

If the process truncates after the Format command line, then it will not work on your system. Unfortunately, the upgrade requires 41 cylinders and will not work with only 40.

We sincerely regret the inconvenience caused by the omission of the warning.

Entering Text from Keyboard with Videotex Plus

Our thanks to Alan Kaplan, CompuServe ID: 73026,637, for this Model III communications hint.

While using Videotext Plus you can enter text or a program directly from the keyboard into the buffer and save it for later transmission. To do this you just change the duplex setting to half-duplex.

First press the up arrow and Q. At the Query Menu select 6 and enter H for half-duplex. (ENTER) (ENTER) will return you to the main screen. Open the buffer (up arrow O) and begin typing. When you have finished your entry, save it to disk (up arrow S and give it a filename). Now you can reload the file when you are ready to transmit.

Communications Corner

by Al and Dru Simon

Ah, springtime! When a young man's fancy turns from thoughts of love to thoughts of how to break into somebody's computer system!

We know of a number of system operators who were very unhappy when the movie WAR GAMES was released because it evidently gave the idea of breaking into computer systems to the few remaining youngsters who hadn't yet thought of it!

This month we'd like to begin talking about security and some of the questions and options one has in the great battle between the sysops and the "hackers".

WHAT'S A HACKER?

A hacker is a person who is gaining unauthorized entry into a system, or breaks any other type of computer security system. The term is also used to describe a person who is so completely immersed in his computer that he never leaves it except for meals, sleep or to read computer magazines.

COULD A HACKER TRULY BREAK INTO A SYSTEM, AS IN "WAR GAMES"?

Yes. Someone armed with even the lowest level password and some knowledge of the structure of the system they wish to break into can get in and bang away at it. By "bang away" we mean setting the intruding computer into a loop which will repeatedly try various combinations of password and secondary level security in order to eventually fall upon a correct code and thus gain entry to the system.

It isn't all that hard to acquire a low level password either, for as long as one person has a password, sooner or later another will have it too. Some will willingly give away or sell their passwords, while others may be discovered written in a wallet or noted on a memo or overheard in a conversation.

IS IT ILLEGAL FOR SOMEONE TO BREAK INTO A SYSTEM?

Yes, it is now a federal offense to break into someone's system. A recent case of some teenagers who went to jail for precisely this "mischief" illustrates the serious light in which "computer piracy" is now viewed.

WHY DO HACKERS WANT TO BREAK INTO A SYSTEM?

There are many reasons. For fun. For stature among peers. People new to computers tend to think that it is a great accomplishment to break into a system and consider it a challenge which they cannot resist.

Some hackers break into systems just to leave their name so the system operator knows that they were there (ala KILROY?).

Sometimes it is done for malicious purposes. We recently were made aware of a "pirate" bulletin board which blatantly advertised several codes owned by various corporations. Callers were invited to break into these corporation's computers and place illegal purchase orders in the names of their worst enemies, thus creating very real trouble for people they don't happen to like.

There are many other reasons why hackers break into computers. The classic one is to raise the sagging grades of a college student or look up someone else's school records or insure getting accepted into a particular class which has limited enrollment etc.

Sometimes corporate computers are broken into in order for the hacker to obtain internal information about the financial state of the company, or to gain access to sensitive material of one sort or another.

HOW DO HACKERS BREAK INTO A SYSTEM?

By "banging away" as we described above, usually aided by an internal accomplice who provides some information about the system. Without the help of someone who is familiar with the system layout and protocol, breaking into a computer would be a tremendous job, for the hacker would have to start from "scratch", trying to figure out what type of system layout any particular company or computer employs, what protocols are being used, and what profit there might be in breaking into the system in the first place.

Usually the hacker begins by obtaining someone's password, then uses it to gain elementary access. Once logged onto the system they "bang away" for as long as they can, trying to obtain a higher security clearance. (In the case of someone illegally ordering merchandise in someone else's name higher security is often not needed at all.)

Once higher clearance is obtained in this manner the hacker is free to peruse any data at that security level within the system, which may include sensitive material, accounting records, personnel records and so forth.

From there they may try to obtain even higher security clearance in the same manner.

WHAT CAN THEY DO ONCE THEY'VE BROKEN IN?

Beside those things mentioned above, some hackers enjoy breaking into bulletin board systems in order pull pranks such as changing parts of the host program, reading private mail, or stealing operational programs. Some hackers find great amusement in breaking out of bulletin board programs into the operating system and remotely formatting all the disks on the system, thus entirely destroying the database. Lots of fun.

IS THERE ANY SYSTEM THAT IS COMPLETELY SAFE?

No. It seems to be human nature that people are willing to allow their passwords to be discovered, and as long as this happens, systems will be broken into.

One can lower the chances by careful programming and making personal accountability a very high priority within business systems but that will not completely eliminate security leaks.

HOW DO YOU LET SOME CALLERS IN (SALESMEN, ETC) WHILE KEEPING OTHERS OUT?

Aside from personal passwords which allow security at only certain levels, there are call-back systems whereby the caller dials into a piece of hardware which looks up his name or clearance in a file, disconnects, and dials him back, thus ensuring that only specific locations get connected, although it cannot ensure that the specific caller on the remote terminal is the person who is supposed to be there.

This method becomes impractical when a company has salesmen travelling to various locations who must call into the computer from any number of locations. The only recourse in this situation is to have these salesmen change their own passwords daily. Another measure in such cases would be to limit the number of logon attempts that can be made before an automatic disconnect, thereby reducing to a minimum the number of attempts that it is physically possible for a hacker to make with an auto dialer.

IS IT TRUE THAT ANY SYSTEM THAT CAN BE PROGRAMMED CAN BE BROKEN INTO?

Pretty much, yes. Many systems have excellent security which will lock out remote input without certain security clearance, but for the most part, SOMEONE will always have such clearance, and that person will have a series of passwords which allow him/her access. As long as there is a password there is a way to discover it.

HOW CAN HACKERS BE PREVENTED FROM "BREAKING AND ENTERING"?

Ah, we knew you'd get around to asking that sooner or later. There are several methods which will reduce the chance of a breach, though none is truly foolproof.

Since hackers often rely on the "banging away" method of discovering entry codes, the computer system can easily thwart this by limiting the number of attempts to access any part of the system. For example it may allow three attempts at entering a code, and assume after the third unsuccessful attempt that the person calling is a hacker. At this point it will automatically terminate the connection. Indeed, the hacker can call back and try again but after a point the expense of the hacker's phone calls will override the profit he expects to make by breaking into the system.

Another security measure is the frequent changing of passwords. This will significantly decrease the chances of a hacker stumbling onto a valid password by the usual "banging away" method.

In the case of businesses which give passwords to individuals, personal accountability is a MUST. If each person with access to the computer has a separate password and security is breached, it is easy to see which password has been used, and the person giving away his code is immediately recognizable and accountable.

Encryptation of code during transmission is an excellent idea as well and can be accomplished easily. We will go into detail about encryptation of code in next month's column.

ARE THERE ANY COMMERCIALLY AVAILABLE SECURITY SYSTEMS?

There are security devices available such as encryptors, callback systems, and "filtering" systems which accept a password for access to logon itself. This just adds one more level of security and can either be done through data or through multi-tones on a telephone.

The above are devices which may be added to a system. We are not currently aware of any independent security "systems" per se.

We will delve further into the subject of computer security in next month's column, and look forward to receiving any further questions you may have regarding this subject.

****

THE NEWS FROM PLUMB

Ric Manning sent us some more news recently and we though you'd like to take a peek:

*Run a BBS without a disk drive? Everyone told Jim Howard it couldn't be done. But Howard, of Raymore, MO, doesn't discourage easily . . . The system is called Howard's Notebook. It uses a TRS-80 Model 1 with 48K of internal storage and a supplemental tape drive. But the system only uses the tape to record log-in data and feedback to the Sysop. Everything else is stored in RAM which means the Notebook can respond faster than a lot of disk-based systems.

The board is very friendly and easy to use. Callers are greeted with a modest menu of 10 numbered options rather than "the alphabet soup that you get on other boards," he said. "My system is ideal for the first-time callers because everything is on the screen. If they make a mistake, they get right back to the menu."

The Notebook is online 24 hours at (816) 331-5868.

*Thanks to a grant from the 3M company, Minnesota physicians have a computerized discussion center where they can exchange ideas on a variety of medical topics. The Minnesota Medical Conference-Tree contains branches on 21 medical specialties, including cardiology, endocrinology, infectious diseases, pediatrics, dermatology, neurology and general surgery. The tree is planted at Cambridge, MN, online around the clock at (612) 434-6315.

*The orchestra section has returned to the Ballroom on The Bread Board System headquarters board in Aurora, CO. Owners of the Orch 80/85/90 programs are invited to sample the music and contribute new songs. The TBBS HQ system also offers TRS-80 user group news, an adventure newsletter, a separate section for the Colorado Ground Water Association and the Sysop is looking for someone to run a CP/M interest section. Report to headquarters at (303) 690-4566.

If you'd like to subscribe to PLUMB or just get in touch with its editor Ric Manning you may do so by writing Riverside Data Inc., P.O. Box 300 Harrods Creek, KY 40027, or calling (502) 228-3820. They also may be reached via CompuServe account 72715,210 or Source account STQ007.

****

THE CORNER MAILBOX

Dear Al & Dru,

I recently purchased the Direct-Connect Modem I (cat 26-1172). In doing so I looked through some past issues of Microcomputer News . . . I read a letter . . . concerning the

procedure of signing on with a Model II to a BBS and I was wondering if you could explain the procedure of signing on with a Color Computer.

I would also be thankful if you would include a list of Bulletin Board Services in my area and ones on a toll free number.

Thank you,
Mark Noel
Sayre, PA

Dear Mark:

First you load your VIDTEX program either by turning off your Coco, plugging in your cartridge and turning the computer on, or by loading your cassette recorder with VIDTEX and then typing SYSTEM and when you get the asterisk prompt type VIDTEX. When you get the next prompt type a slash (/) then press the ENTER key. If you're on disk, type LOAD VIDTEX.

Once the program has loaded into the Coco you'll see a header. Press the BREAK key. After contact has been established with the BBS press the ENTER key. Once you've logged onto a BBS it's a matter of whose board you're logging onto. If you see no printing, keep on pressing the ENTER key until you get a response. If you get no response the board is probably down, or requires a control code to be entered at that point.

Take it from there. Make sure that the switch on your modem is set in the proper position for your computer (refer to your manual).

Although we're not aware of any boards which can afford a toll free number, we received the following letter at just about the same time we received your letter. Take a look!

* *

Dear Sirs:

Please find enclosed a partial list of the Color-80 BBS systems that are operational across the country as of this time. All of these systems are running on 64K TRS-80 Color Computers with two drives or more.

Thank you!
Shawn Jipp
Sunnyvale, CA

Alberta, Canada	(403) 474-0147
Arlington, MA	(617) 646-6809
Atlanta, GA	(404) 378-4410
Canfield, OH	(216) 788-7910
Canton, MI	(313) 981-5061
Chicago, IL	(312) 397-8308
Ft Wainwrt, AK	(907) 356-COCO
Greenville, NC	(919) 758-5261
Henderson, TX	(214) 657-8147
Highland Pk, NJ	(201) 572-061
Hixson, TN	(615) 842-68
Kansas City, MO	(816) 358-62
Lawton, OK	(405) 248-8433
Minneapolis, MN	(612) 533-1957
Montrose, CO	(303) 249-7866
Morgantown, WV	(304) 599-0760
Mt. Lk Terr., WA	(206) 481-6549
Parkesburg, PA	(215) 857-3035
Pensacola, FL	(904) 456-7195
Phoenix, AZ	(602) 245-0488
Portland, OR	(503) 761-6345
Rochester, NY	(716) 381-6800
Toronto, Ontario	(416) 767-0412
Travis AFB, CA	(707) 437-6336
Upper Mrlbro, MD	(301) 599-1726
Wallingford, CT	(203) 237-2668

PLEASE NOTE: Because of the obvious length of this list we have omitted all but one representative board from each state. We feel certain that each board contains the entire list of its members in the net and apologize to the sysops of those boards which were not listed here.

* *

Hi:

We purchased a TRS-80 Model 12 to use for business and personal use . . . we are having trouble finding (programs) for our kids to gain computer experience on. Can programs for Models I,III, IV or Color be used on our model 12 somehow? HELP! Please don't suggest a computer class as we are somewhat isolated, it is a 2-3 hour trip to the nearest town large enough to have a computer center or offer classes.

Ed Kendrick
Gooding, ID

Dear Ed:

Almost all of your basic programs should run on the Model 12 except for those which include PEEK and POKE statements. Most color computer MACHINE LANGUAGE programs will not work on your machine, though BASIC programs may after some editing. CP/M should transfer directly. Relocatable machine language code files (those programs which do not call upon any external subroutines for example a call to TRSDOS or a ROM location) will run well.

All other programs probably will not run. ASCII files can be transmitted to your machine (from say, a BBS). Any program you move from one machine to another must be stored in ASCII. The easiest way to accomplish a transfer is by backing up one machine to another using or making a null modem (26-1496) (we discuss the making of such devices in the June '83 issue) and sending the files directly from one machine to the other. This can be accomplished at a very high baud rate, though if the transfer is to be made over the telephone it will take longer, as the baud rates will probably be limited to either 300 or 1200.

* *

Next month we'll be discussing data encryption, as we as other interesting methods for protecting computer systems from security breaches.

Once more we'd like to thank you for your support and letters. Until next month, HAPPY COMMUNICATING!

Australians Choose U.S. AgriStar System for National AG Network

By Richard Weening

(CANBERRA, AUSTRALIA) AgriData Network (AgriStar), the U.S. electronic information, communications and computing service for agriculture has been chosen as the system for an Australian national agricultural network to be introduced this summer. The planned multi-million dollar project will be the country's first commercial electronic information service for agriculture. AgriData Network is the first U.S. system to be adopted for wide-spread use in Australia, where British Prestel is dominant.

The selection of AgriData was announced January 24th at the Australian National Ag Outlook Conference by Mr. Peter Nixon, former Minister of Primary Industry (Secretary of Agriculture). Nixon, who is now Chairman of Techcom Australia Pty. Ltd., said the new service would also be known as AgriData Network. Nixon, a farmer and holder of several cabinet level posts in the Australian government, will serve as Chairman of AgriData Australia.

"The AgriData Network U.S. is the most advanced and comprehensive information system and communications service in its field in the world," Nixon said. "That's the plain conclusion we have reached after extensive evaluations of systems around the world," he concluded.

The systems evaluated included 12 U.S. on-line or ASCII based systems and 12 videotex systems, including European, Canadian, and U.S. technologies.

Richard Weening, AgriData CEO, said, "Being chosen as the system for an entire country, particularly Australia, a major player in world agriculture, is like an official stamp of approval."

Nixon said the AgriData Network Australia will be extensively modified "to serve the needs of our country and it will be satellite-linked to the U.S. system for a continuous transfer of information in both directions."

AgriData Network is operated by AgriData Resources, Inc., the Milwaukee-based business information publishers for agriculture. AgriData publishes "FarmFutures", a 200,000 circulation monthly national business magazine for large farmers and ranchers. The company also publishes a range of weekly newsletters and market advisory services.

AgriData Network (AgriStar) is distributed nationwide by Radio Shack. It is available for free demonstration in Radio Shack computer centers, stores and dealers throughout the U.S. and Canada.

The system offers users with a data terminal or microcomputer on-demand access to a wide range of continuously updated news, market price quotations, weather and general business, financial and economic data. Over 300 different sources of data feed into the system from around the world including Commodity News Service (CNS), the Associated Press (AP), and the various commodity exchanges. The system already serves more than 10,000 terminals throughout the U.S. and Canada including the Secretary of Agriculture and the House and Senate Ag Committees.

The system is completely compatible with any TRS-80 micro-computer. All you need is a 300 or 1200 baud modem. When you purchase the service from Radio Shack, you get free communications software that automatically connects your TRS-80 over phone lines to the system.

AgriData Network is widely regarded as the finest service of its type available anywhere. The system is used to manage information which is external to the agricultural business. Together with Radio Shack/Ag Disk software for internal information management like accounting, you have a power tool as important to farm profits as any implement you own.

AgriData Network (AgriStar) requires no understanding of software or computers. The system responds fully to English language commands and takes less than 15 minutes to master.

Electronic mail service is available through the network's StarGram system. You can even use the network to send mail messages with a guaranteed 48-hour delivery to any postal patron in the continental U.S. for $1.00.

So, if you are looking for a powerful and profitable new use for your micro, get a free demonstration at your nearest computer center, store or dealer.

What about cost? The Basic package, "AgriData" (Cat. No. 26-2227), costs only $199.95 , entitling you to 6 months of the Basic service, after which you pay $39 per month. In addition, you pay $25.00 per hour when connected to the system which is so fast that most sessions are less than 5 minutes. A nominal charge averaging $.25 is made for each report you access. The bottom line—including your initial charge, the average monthly cost is about $80 for information which could save or make you thousands in marketing prices achieved, help reduce the risks of price savings and more.

Best news of all—use of the service is toll-free. You pay no long distance charges from anywhere in the U.S. The Australians chose it as the best system in the world after 15 months of study. Take a few minutes and see why.

Pascal File Dump Utility Program

by Earl W. Bollinger

When you are developing computer programs for the Model 2000, it is sometimes useful to be able to examine a file in detail. I developed the Pascal File Dump Utility program for just that purpose. Although the MSDOS debugger utility is able to perform this task, I sometimes find it convenient to be able to dump files without invoking the debugger. This program was written using Microsoft's MS-PASCAL compiler, version 3.13.

(Editor's Note: In addition to the Pascal source code, we have provided some of the information from the listing which we thought might be helpful.)

Source Code

```
program dump(input,output);
{ dump -- displays the file entered as hexadecimal
     ASCII format data }
{   on the standard output device.  }
{ BY E.W.Bollinger on Oct. 14, 1983   ver 1.Ø }
VAR
  f : packed file of char;
  c : char;
  filename : string(4Ø);
  linestr : packed array[1..72] of char;
  astr : packed array[1..5] of char;
  bstr : packed array[1..2] of char;
  addr, maximum : integer4;
  charcount, linecount, i, ch, offset : integer;
  done : boolean;

function hexchar(n: integer): char;
{ hexchar -- converts number to a hex char value }
var
  c : char;
begin
  case n of
    Ø: c:='Ø';
    1: c:='1';
    2: c:='2';
    3: c:='3';
    4: c:='4';
    5: c:='5';
    6: c:='6';
    7: c:='7';
    8: c:='8';
    9: c:='9';
   1Ø: c:='A';
   11: c:='B';
   12: c:='C';
   13: c:='D';
   14: c:='E';
   15: c:='F'
  otherwise
       c:='Ø'
   end;
  hexchar:=c;
end; { of hexchar }

procedure hexit(num: integer4);


{ hexit -- converts integer4 number into hexadecimal
     string }
var
  h,temp,setval,maxval : integer4;
  x : real;
  val : integer;
begin
  maxval:=1Ø48575;
  setval:=65536;
  h:=num;
  temp:=num;
  val:=Ø;
  num:=h div 65535;
  if num>Ø then
   begin
     x:=float4(num);
     val:=round(x)
   end;
  while h>65535 do
       h:=h-setval;
  astr[1]:=hexchar(val);
  if temp=65535 then astr[1]:='Ø';
  if temp=maxval then astr[1]:='F';
  val:=Ø;
  while h>4Ø95 do
    begin
      h:=h-4Ø96;
      val:=succ(val)
    end;
  astr[2]:=hexchar(val);
  val:=Ø;
  while h>255 do
    begin
      h:=h-256;
      val:=succ(val)
    end;
  astr[3]:=hexchar(val);
  val:=Ø;
  while h>15 do
    begin
      h:=h-16;
      val:=succ(val)
    end;
  astr[4]:=hexchar(val);
  val:=Ø;
  if h>Ø then
   begin
     x:=float4(h);
     val:=round(x)
   end;
  astr[5]:=hexchar(val);
end; { of hexit }


procedure hexbyte(c: char);
{ hexbyte -- convert a byte to hexadecimal notation }
var
  h,val : integer;
begin
  h:=ord(c);
  val:=Ø;
  while h>15 do
    begin
      h:=h-16;
      val:=succ(val)
    end;
  bstr[1]:=hexchar(val);
  bstr[2]:=hexchar(h);
end; { of hexbyte }

procedure header;
{ header -- used to output a dump file screen header
     line }
begin
  writeln;
  write(' ADDR  Ø  1  2  3  4  5  6  7  8  9  A  B
     D  E  F');
  writeln('   Ø123456789ABCDEF');
  writeln;
end;
```

```
BEGIN { MAIN PROGRAM }

writeln;
writeln;
writeln('     FILE DUMP UTILITY PROGRAM    ver 1.Ø');
writeln;
writeln;
write('Enter Filename: ');
readln(filename);
writeln;

assign(f,filename);

reset(f);

maximum:=1Ø48476;
addr:=Ø;
charcount:=Ø;
linecount:=Ø;
done:=false;
 offset:=7;
 for i:=1 to 72 do
     linestr[i]:=' ';

header;

hexit(addr);
for i:=1 to 5 do
    linestr[i]:=astr[i];

while not(done) do
 begin
   if eof(f) then
      done:=true
   else
     begin
       read(f,c);
       addr:=addr+1;
       if addr=(maximum) then
          addr:=Ø;
       hexbyte(c);
       linestr[offset]:=bstr[1];
       linestr[offset+1]:=bstr[2];
       offset:=offset+3;
       charcount:=charcount+1;
       ch:=ord(c);
       if (ch>31) and (ch<128) then
          linestr[charcount+56]:=c
       else
          linestr[charcount+56]:='.';
       if charcount=16 then
         begin
           for i:=1 to 72 do
               write(linestr[i]);
           writeln;
           for i:=1 to 72 do
               linestr[i]:=' ';
           offset:=7;
           linecount:=linecount+1;
           if linecount=21 then
             begin
               header;
               linecount:=Ø
             end;
           charcount:=Ø;
           hexit(addr);
           for i:=1 to 5 do
               linestr[i]:=astr[i];
         end;
       end;
     end; { of while }
   for i:=1 to 72 do
      write(linestr[i]);
   writeln;
   writeln;

   close(f);
end. { of dump program }
```

Listing

```
Symtab     42   Offset Length   Variable - HEXCHAR
                -    2      6   Return offset, Frame length
                -    2      1   (function return)              :Char
                +    4      2   N                              :Integer ValueP
                -    4      1   C                              :Char

Symtab     95   Offset Length   Variable - HEXIT
                -    4     24   Return offset, Frame length
                +    4      4   NUM                            :Intger4 ValueP
                -    4      4   H                              :Intger4
                -   2Ø      4   X                              :Real
                -    8      4   TEMP                           :Intger4
                -   22      2   VAL                            :Integer
                -   12      4   SETVAL                         :Intger4
                -   16      4   MAXVAL                         :Intger4


Symtab    112   Offset Length   Variable - HEXBYTE
                -    2      6   Return offset, Frame length
                +    4      1   C                              :Char    ValueP
                -    2      2   H                              :Integer
                -    4      2   VAL                            :Integer


Symtab    121   Offset Length   Variable - HEADER
                -    Ø      2   Return offset, Frame length
```

```
Symtab   2ØØ   Offset Length   Variable
                    Ø    798   Return offset, Frame length
                   2Ø    636   F                          :File    Static
                  656      1   C                          :Char    Static
                  658     4Ø   FILENAME                   :Array   Static
                  698     72   LINESTR                    :Array   Static
                  77Ø      6   ASTR                       :Array   Static
                  778      4   ADDR                       :Intger4 Static
                  79Ø      2   I                          :Integer Static
                  792      2   CH                         :Integer Static
                  776      2   BSTR                       :Array   Static
                  794      2   OFFSET                     :Integer Static
                  796      1   DONE                       :Boolean Static
                  782      4   MAXIMUM                    :Intger4 Static
                  786      2   CHARCOUNT                  :Integer Static
                  788      2   LINECOUNT                  :Integer Static
               Errors  Warns   In Pass One
                    Ø      Ø
```

Express Order Software

The Express Order Software program is a new and innovative method of marketing software.

The basic concept of this program is to make market-proven, high quality software from third party (non-Radio Shack) vendors available to the public through the Radio Shack store distribution network.

Customers are advised of software availability through in-store product descriptions, which are updated frequently. A quarterly "carry out" catalog, describing the Express Order Software program and the available software, is also planned.

Customers can go to any Radio Shack store, Radio Shack Computer Center, or participating dealer nationwide to place an Express Order Software order. Orders are electronically processed the same day. The software is stocked in Tandy Corporation's main warehouse in Fort Worth, Texas, and is shipped the next day to the store for delivery to the customer.

Software in the Express Order Software program will be:

- initially, MS-DOS based generic applications, such as word processors, data bases, etc. In the future, special application software that supports other TRS-80 computers will be considered and may be added to the Express Order Software product line, and
- market proven. In most cases it will already have established a name for itself in the marketplace.

Participating vendors will provide:

- complete packaging in shrink wrapped form, and
- complete customer support to the end-user, including a telephone number on the outside of the package. This level of support is consistent with the support currently provided by Radio Shack to its customers.

Software vendors interested in the Express Order Software program should submit a letter of interest to:

Express Order Software
1300 One Tandy Center
Fort Worth, TX 76102

Customers interested in the Express Order Software program should contact their local Radio Shack Dealer.

Magazines

Below are nine magazines of special interest to TRS-80 owners that we believe have editorial content of high quality and will be of use to our customers.

Basic Computing—The TRS-80 User Journal
3838 South Warner Street
Tacoma, WA 98409
(206)475-2219

Color Computer Magazine
Highland Hill
Camden, ME 04843
(207)236-9621

Computer User
16704 Marquardt Ave.
Cerritos, CA 90701

80 Micro
P.O. Box 981
Farmingdale, NY 11737

Hot CoCo
P.O. Box 975
Farmingdale, NY 11737

Portable 100—The Magazine for Model 100 Users
P.O. Box 468
Hasbrouck Heights, NJ 07604

PCM—The Portable Computing Magazine
9529 U.S. Highway 42
P.O. Box 209
Prospect, KY 40059

Rainbow (Covers the TRS-80 Color Computer)
P.O. Box 209
Prospect KY 40059
(502)228-4492

two/sixteen magazine
P.O. Box 1216
Lancaster, PA 17603
(717)397-3364

Musical Notes

by Bryan Eggers
Software Affair

Hardly a day goes by without someone finding a completely new use for a well-known product. I'm sure you'll agree, for example, that a doorknob is a great place to hang a coat, especially with a closet full of tangled hangers!

We're constantly surprised at the new techniques developed for the Orchestra-90™, too. Users are developing all kinds of special effects and arranging tricks. Some amazingly complex arrangements have been submitted to our SIG database.

Most of these new programming ideas appear in the message section of our ORCH-90 SIG on CompuServe. Users often share their new "tricks" with us, but since these messages eventually scroll off and disappear we felt that a permanent storage area was necessary.

We created a new section on CompuServe called the ORCH-90 ARCHIVES. Typing GO ORC-1 from any CIS page prompt will get you to the top menu of the ARCHIVES.

The ORCH-90 ARCHIVES allow us to permanently store these text files, then users can easily access them by following standard menu-driven prompts. Anyone may contribute an article to this area by submitting it invisible to Public Access, then notifying Software Affair on our ORCH-90 SIG (page HOM-13) when it becomes available. We'll insert the special page formatting commands and merge your file into the ARCHIVES.

I'm reproducing some of the articles for you this month, but I'll never be able to print all the new information in this limited space. So if you've been considering buying a modem, now is the time to do it! Once you get a CompuServe account, membership in the ORCH-90 SIG is free!

If you have any questions, suggestions or programming tips you'd like to share with other readers, please send them to my attention c/o TRS-80 Microcomputer News.

Here are a few of the articles in the ORCH-90 ARCHIVES:

PROGRAMMING TRICKS

By Larry Alexander

Here's a technique whereby one voice appears to play chords. One voice plays all the notes of the chord rapidly and fools the ear into thinking more than one music voice is involved.

Load your ORCH-90 program, go into Edit mode and type in the following short music file:

```
NH=7Ø V1YA KØ& <1
PØ1
MØ1 *S(A616)3
RØ1 RØ1 RØ1
```

The single voice in measure 01 is playing the three notes so rapidly that it sounds like a full chord.

This time we'll use Voice 1 to play a melody, adding Voices 2 and 3 for some harmony, and using Voice 4 to simulate the same "chord" effect:

```
NH=7Ø KØ& <1 Z3
V1YA V2YA V3YA V4YE
PØ1
MØ1 *SBI.C"SAI.B"S9I.A"S8I.9"
V2 *(Q$S6D")1
V3 *(Q$S3A")1
V4 *S(A616)3
RØ1 RØ1 RØ1
```

This produces an effect of more than four voices playing and creates a very full background accompaniment for the melody.

You may wish to download my arrangement of "Old Man River" from the SIG database and see how I use this one-voice "chord" effect throughout the arrangement. The filename is "OLDMAN.A85" and it's in the XA2 database.

Listen to the section of "Old Man River" where the bass carries the melody. Playing rapidly shifting octaves with one voice like this can create the effect of TWO voices! Try it. To the ear it sounds like the melody is being played by two separate voices even though it is actually just ONE voice! You can even do this with all four voices simultaneously . . . giving the effect of EIGHT voices! With ORCH-90 on a Model 4 you can simulate the effect of TEN voices playing at the same time!

Don't hesitate to use humor, alternate harmony and sheer "nuttiness" in your arrangements and transcriptions as I did in my arrangement of "Alexander's Ragtime Band". The old "ricky-tick" sound was intentional! Lighten up! Make your music fun to listen to. I don't transcribe music . . . I write all arrangements "from scratch", but you can make your transcriptions from sheet music more interesting by not entering them simply "note for note". Vary the notes and rhythm patterns here and there. Then play it back again until it sounds good to you. There is no hard, fast rule that says you must enter each note of your transcription EXACTLY as it appears on the sheet music. If your final product really sounds good to you, then it will more than likely sound good to everyone else who hears it!

CHOOSING THE TUNE

by Jerry Bradshaw

Of all the considerations in arranging, whether it be for the Orchestra-90 system or for a live group, this may well be the most important question to consider:

"How do I select a tune to arrange?".

A good feeling for this question and an understanding of the rationale behind the answers will go a long way towards making you a better arranger. I hasten to add that the thoughts contained in this monograph are my own, compiled

over some 26 years of arranging experience for groups ranging from small vocal ensembles to full band and orchestral arrangements and are intended to be no more than a guide to those interested in knowing how I go about it.

For many people, choosing a tune to arrange is a simple process. These people have a strong interest in a particular musical style or in some favorite musical group or soloist and will often devote many hours to arranging the complete catalog of everything their favorites ever sang or played or as many pieces in that style as they can find. Quality is not the keyword here but quantity is important to these people. I suppose we all fall into this trap at some time or other but, while this sort of activity does tend to keep us off the streets, it certainly cannot be the most constructive use of our time.

Most tunes done by any group are, at best, only modestly successful while a very few are truly outstanding. When most people purchase a new album they will usually find only one or two tunes from that album which will really justify the purchase of the album and the rest of the album will then fade into relative obscurity. So, the point arises as to the need to arrange tunes which may have little merit to begin with to the average listener while the truly outstanding tunes are left either undone or to the pens of the more experienced, advanced arrangers.

When I'm looking for a new tune to arrange for the ORCH-90 there are several questions which I usually try to consider. With so many new tunes hitting the market every day and the wealth of the existing library of music in the world, I often find myself having great difficulty settling on what to work on next!

The first question I ask is:

Is it musical?

This may well be the most important question because if the tune isn't very musical to begin with, who's going to want to listen to it? Also, I have found that the less musical the tune is to begin with, the more difficult it is to arrange. It's a lot harder to make a silk purse out of a sow's ear than it is to make it from silk!

Secondly, I ask myself:

Do I like it?

If I don't really like a tune, no matter how great a tune it is, I won't arrange it because I won't really feel like contributing to the development of the tune within the framework limitations of the ORCH-90 system. I have to feel an empathy with the composer and what he/she was trying to communicate to the world. If I can't have that feeling, I have a great deal of difficulty doing the tune!

Third, I ask:

How will it sound on the ORCH-90?

A lot of really good music just doesn't sound good on the ORCH-90 system. (I hasten to add that frequently the converse of that statement is also true in that many tunes may well sound better on the ORCH-90 than in their original setting!) Taking into consideration all of the features and means of enhancement available within the ORCH-90 system, one has to decide if these features can be used to their best intent on the tune in question. This implies a judicious and intelligent use of the features available.

Fourth, ask yourself:

Is this a tune that I would want to show off to not only my friends but also to my critics?

Public acceptance of an arrangement has a degree of importance which is not easily measured. Many users (who are also non-arrangers) will eagerly snatch up every new tune which comes from the computer of certain arrangers because they know that the work can well be shown off to friends without being an embarrassing bore! Notice what kinds of tunes these arrangers usually choose!

I'm certain that different people will have different ideas as to how to go about choosing a tune and that's fine! Whatever works for you is your cup of tea. Hopefully, these thoughts may be of some assistance in developing a means of setting standards of what music will eventually wend its way through the chips and gates of all of our faithful TRS-80's.

VOICE TRANSPOSITION VALUES

by David Plumlee

Surely you have seen U7, U0, and U + 7 often in ORCH files. But, what about other values such as U4, U6, or UE ? Well, the manual said that any hex digit would work, so I decided to give it a whirl.

I had the tune 'MARJORIE' that has been on this SIG, and it became my experimental piece. Also as a lark, I wanted to prove how much I could change the sound of a piece without changing a single note.

After changing some registers, I applied some U values to different voices. As I remember, I ended up with a U7 in V2, a UE in V5, and a U5 in V3. If you want to hear the results of my fun and games, download "MARJR2.A85" from the XA2 database. The sound is a bit strange, but who knows, it may give someone an idea of what can be done with U values other than 0 or 7.

Incidentally, for the new ORCH user, let's briefly examine U, < and >. You know that U affects a single voice where < or > affect all voices (the entire piece). More important, to move an octave with U requires U-7 to lower the voice one octave (U + 7 will raise the voice one octave), whereas to raise or lower the entire piece a whole octave requires a >C or <C, respectively.

When you use a U value, the voice respects your key signature. Thus my V3 of MARJR2.A85 sounds MAJOR where the other voices were MINOR. My U value gave me the relative major of the original. On the other hand, < slides everything down chromatically by the number after it. In "REJOIC.A85" I cheated on the Bach part: first, the theme appeared in the key of G, then in D. Using a word processor, I block duplicated a copy of the original part. On this second part I added a <5. Eureka! I got out of writing about 20 measures of music. So don't be afraid to play with U and < > values. You may find interesting sounds.

BASS VOICING

by Larry Alexander

The register definition JESFF8FFF0FF is perhaps the best bass sound that can be obtained with ORCH-90 (personal opinion). It works very well with all types of music. ORCH-90 nicely supports bass lines an octave lower than written (U7) as long as the voice isn't written too low.

For a really strong bass voice, try "tracking" the bass line with another voice playing every note a fifth above. I used this effect in "DO-RE-MI" (DOREMI.A85). Another way to get a strong bass line is to cut the value of each bass note in half and write the second note of the pair an octave higher.

Instead of this:

```
V5@QE"D"C"B"A"B"C"D"WE"
```

do this:

```
V5@IE"7,D"6,C"5,B"4,A"3,B"4,C"5,D"6,WE"
```

Play the above example at a fairly fast tempo for best demonstration of the effect.

"FUZZ" BASS

by Larry Alexander

You can achieve a pretty good "fuzz" guitar or "fuzz" bass effect by setting a voice to the percussion register and playing music. Be sure to stay within the note limits of 0 to 9 and A to F for best results. Download "I've Got a Woman" from the XA2 database for an example of this effect. The filename is WOMAN.A85.

COMPILER TRICKS

by Jon Bokelman

I've noticed a few music files in the XA2 database that make extensive use of a doubled bass line. That is, identical notes played in two voices, for example:

```
V3 @Q34HØ
V4 @Q34HØ
```

Using reiterative compilation, the data entry and the size of the piece can be reduced to:

```
V3(@Q34HØV4)1
```

The result is Q34H0 being compiled into V3 and then nto V4.

The reiteration process is totally dumb; any symbols, except numbered Parts, can appear between the parenthesis. Experiment!

RESTS

by Larry Alexander

The use of many rests in your music is a waste of valuable voice space. Instead of writing a rest in a voice, put in a bass clef sign and have that voice double the bass voice for a few notes. Or have the voice double the lead voice. Judicious use of this technique can make your music files sound bigger and fuller. For continuity, you need to use this technique as consistently as possible throughout your file. Try it!

Getting Extra Disk Space with Profile III +

Daniel Sapp
Director Special Education
L'Anse Creuse Public Schools
47260 Sugarbush Road
Mount Clemens, MI 48045

In the past year our Special Education Department has become a frequent user of the Profile III + program for our student registry, building budget, and an unending list of other uses. However, adequate disk storage space has been a problem. The "System" requirements of the Runtime disk severely limited the size and type of file that we could use on a two disk drive system.

This summer, I discovered a method for gaining the extra disk space needed. Briefly stated, both the Runtime and Creation "CMD" are placed on a single disk to be used in drive <0> and a blank formatted disk is used in drive <1>. Each time the program asks for the drive number that contains the Runtime disk you must enter the number <1>.

If a merge with SCRIPSIT or VISICALC is desired, copy the Runtime "CMD" files onto the SCRIPSIT or VISICALC disk. Once the Runtime files are on the appropriate disk, record selection, document creation, and the merge can take place using the "DATA" disk in drive <1> and the "PROGRAM" disk in drive <0>.

This method of using the Profile III + program provided approximately 233 free granules for program and data storage on a single disk and significantly improved the usefulness of the program for our needs. For your convenience I have included a brief, more complete, description of the process.

PROCEDURE

1. Insert Profile III + Creation disk in Drive 0 and the Runtime disk in Drive 1.
2. At "TRSDOS READY . . ." type "COPY /CMD:0 :1" and press ENTER. This will copy all the Profile Creation programs onto the Profile Runtime disk.
3. Remove the Profile Creation/Runtime disk from Drive 1 and put it into Drive 0.
4. Insert a blank "DATA" disk in Drive 1 and at "TRSDOS READY . . ." type "FORMAT :1" and press ENTER.
5. You may now set up your Profile programs. Remember always answer the question "WHICH DRIVE HAS THE RUNTIME DISK?" with Drive 1. This will put all your files on your formatted data disk in Drive 1. You will always need a Runtime Disk in Drive 0 to operate your programs. However, you will generally have in excess of 200 free granules for file expansion or about 500 large records.

Bulletin Boards

COLORAMA

A bulletin board run by The Color Force computer club. President of the club is Emery Mandel. If you have questions or would like to join The Color Force, you can call Reid Baker at (813) 733-5095 or call the BBS at (813) 733-2415. We prefer you call voice. (For the mailing address, see The Color Force in the Computer Clubs column.)

CAPATUG BBS

A bulletin board run by the CAPATUG Users Group. The bulletin board number is: (717) 774-6543. (For the mailing address, see CAPATUG in the Computer Clubs column.)

Inventory/Records

Tom Trepanier
35 Lincoln Ave.
Ardsley, NY 10502

I am 14 years old and have recently written a home inventory program which I call "Inventry" and a program called "Records" for the Model I/III/4. Both programs create a DATA FILE and allow the operator to ENTER data, LOOK UP a record, SCAN ALL RECORDS on file, UPDATE a record, DELETE a record, INITIALIZE a file, and EXIT the program.

After you have entered the program, checked it for errors and saved it to disk, you must initialize the file before you can begin entering data. To do this, load the program and at the menu prompt type "I". The screen will display the message "THIS WILL ERASE ALL PREVIOUS ENTRIES, IF ANY! TO CONTINUE INITIALIZATION, HIT THE C KEY". Hit the C and you will see the message "THIS WILL TAKE A LITTLE WHILE". When initialization has been completed you will be returned to the menu and can begin making entries.

The inventory program lets you enter OBJECT, SERIAL #, LOCATION, and COLOR for each item you list. The records program is very similar but the entries are: NAME, ADDRESS, CITY, STATE & ZIP CODE, DATE OF BIRTH, DOCTOR, SOCIAL SECURITY #, MEDICARE, MEDICAID, NEXT OF KIN, TELEPHONE, RELIGION, MEDICATION, SEX, ROOM #, and DIET. In each program, after the last entry per item you see a prompt which reads: ARE YOU FINISHED (Y/N). By responding "N" you can continue entering items. When you have completed your entries simply respond "Y" and the data file will be updated.

Inventory

```
1 REM ** PROGRAM NAME:INVENTRY/BAS **
2 REM ** DATA FILE NAME:INVENTRY/DAT:Ø **
3 REM ** DATA FILE IS ON DRIVEØ **
4 REM ** MAXIMUM FILE SIZE IS 1ØØØ RECORDS **
5 REM ** RECORD LENGTH IS 8Ø PACKED 3 PER SECTOR **
6 REM
7 REM ** BY TOM TREPANIER **
8 REM ** 35 LINCOLN AVE **
9 REM ** ARDSLEY, NY 1Ø5Ø2 **
1Ø REM
11 REM CHANGE DISKS--REINITIALIZE HERE
2Ø CLEAR 3ØØØ
3Ø OPEN "R",1,"INVENTRY/DAT:Ø"
4Ø ON ERROR GOTO 25ØØ1
5Ø DIMF$(22),G$(22)
6Ø FIELD 1, 2ØAS F$(1), 2ØAS F$(2), 2ØAS F$(3),
     2ØAS F$(4)
7Ø FOR I=1 TO 22
     : G$(I)=""
     : NEXT
     : FC=Ø
     : UF=Ø
     : G$=""
     : CLS
8Ø PRINT TAB(18);"PRODUCED BY TOM TREPANIER"
85 PRINT
9Ø PRINT "                    ** HOME INVENTORY **"
95 PRINT
97 PRINT
1ØØ PRINT
11Ø PRINT "ENTER DATA.....DEPRESS E"
12Ø PRINT "LOOK UP A RECORD.....DEPRESS L"
13Ø PRINT "SCAN ALL RECORDS ON FILE.....DEPRESS S"
14Ø PRINT "UPDATE A RECORD.....DEPRESS U"
15Ø PRINT "DELETE A RECORD.....DEPRESS D"
16Ø PRINT "INITIALIZE THE FILE.....DEPRESS I"
17Ø PRINT "EXIT THE PROGRAM.....DEPRESS X"
18Ø PRINT "PLEASE DEPRESS THE LETTER OF YOUR CHOICE
     : NO ENTER KEY IS NEEDED"
19Ø AN$=INKEY$
     : IF AN$="" THEN 19Ø
2ØØ IF AN$<>"E" AND AN$<>"L" AND AN$<>"U"
     AND AN$<>"D" AND AN$<>"I" AND AN$<>"X"
     AND AN$<>"S" THEN 6Ø
21Ø IF AN$="E" THEN 1ØØØ
22Ø IF AN$="L" THEN 1ØØØØ
23Ø IF AN$="S" THEN 35ØØØ
24Ø IF AN$="U" THEN 11ØØØ
25Ø IF AN$="D" THEN 12ØØØ
26Ø IF AN$="I" THEN 32ØØØ
27Ø CLOSE
     : NEW
1ØØØ CLS
     : REM ** BEGIN ENTRY **
1ØØ5 FC=FC+1
1ØØ6 ON FC GOSUB 1Ø1Ø, 1Ø7Ø, 112Ø, 117Ø
1ØØ7 IF FC< 4 THEN 1ØØ5 ELSE 122Ø
1Ø1Ø PRINT "WHAT IS THE OBJECT?"
1Ø2Ø LINE INPUT G$(1)
1Ø3Ø IF LEN(G$(1))>2Ø THEN PRINT "TOO LONG"
     : GOTO 1Ø1Ø
1Ø4Ø IF LEN(G$(1))=Ø THEN PRINT "YOU MUST ENTER
     OBJECT"
     : GOTO 1Ø1Ø
1Ø6Ø RETURN
1Ø7Ø PRINT "WHAT IS THE SERIAL # ?"
1Ø71 IF UF<>Ø AND G$="" THEN G$=G$(2)
1Ø8Ø LINE INPUT G$(2)
1Ø9Ø IF LEN(G$(2))=Ø THEN PRINT "YOU MUST ENTER
     SERIAL #"
     : GOTO 1Ø7Ø
111Ø RETURN
112Ø PRINT "WHAT IS THE LOCATION ?"
113Ø LINE INPUT G$(3)
114Ø IF LEN(G$(3))=Ø THEN PRINT "YOU MUST ENTER
     LOCATION"
     : GOTO 112Ø
116Ø RETURN
117Ø PRINT "WHAT IS THE COLOR ?"
118Ø LINE INPUT G$(4)
119Ø IF LEN(G$(4))=Ø THEN PRINT "YOU MUST ENTER
     COLOR"
     : GOTO 117Ø
121Ø RETURN
122Ø ZZ$=G$(1)
     : GOSUB 26ØØØ
123Ø REM ** LOOK FOR RECORD SPACE **
124Ø GOSUB 125Ø
     : GOTO 13ØØ
125Ø RP=RP+1
     : IF RP> 1ØØØ THEN RP=1
126Ø R%=(RP-1)/ 3 +1
     : P%=(RP-1)-(R%-1)* 3
1265 FIELD 1, 8Ø *P% AS DX$, 8Ø AS ZY$
127Ø GET 1,R%
     : IF ZY$<CHR$(25Ø) THEN 125Ø
128Ø FIELD 1, 8Ø*P% AS TX$, 2ØAS F$(1), 2ØAS F$(2),
     2ØAS F$(3), 2ØAS F$(4)
129Ø RETURN
13ØØ FOR K=1 TO 4
131Ø LSET F$(K)=G$(K)
     : NEXT
     : PUT 1,R%
132Ø PRINT "ARE YOU FINISHED (Y/N)"
133Ø TM$=INKEY$
     : IF TM$="" THEN 133Ø ELSE PRINT TM$
134Ø REM ** IF DONE, END
     : IF NOT, RETURN TO MENU, ELSE REPEAT **
```

```
1350 IF TM$<>"Y" AND TM$<>"N" THEN PRINT "PLEASE
     ANSWER "; "Y"; " OR "; "N"
     : GOTO 1320
1360 IF TM$="Y" THEN 60
1370 FOR I=1 TO 22
     : G$(I)=""
     : NEXT
     : FC=0
     : UF=0
     : CLS
     : GOTO 1000
10000 REM ** BEGIN THE FILE LOOK-UP ROUTINE **
10010 CLS
     : GOSUB 27000        'TRY TO FIND THE RECORD
10199 REM ** UNPACK FIELDS IN FILE FOR DISPLAY **
10200 GOSUB 28000
10799 REM ** DISPLAY RECORD IF MATCH ON KEY **
10800 GOSUB 29000
10860 GOTO 60
11000 REM ** BEGIN THE FILE UPDATE ROUTINE **
11010 CLS
     : GOSUB 27000         'TRY TO FIND THE RECORD
11199 REM ** UNPACK FIELDS IN FILE FOR DISPLAY **
11200 GOSUB 28000
11799 REM ** DISPLAY RECORD IF MATCH ON KEY **
11800 GOSUB 29000
11810 PRINT "WHAT FIELD NUMBER DO YOU WANT TO
     UPDATE?"
11820 INPUT UF
11840 IF UF> 4 OR UF<1 THEN PRINT "INVALID FIELD"
     : GOTO 11810
11860 ON UF GOSUB 1010, 1070, 1120, 1170
11865 R%=(RP-1)/ 3 +1
     : P%=RP-1-(R%-1)* 3
11870 IF UF<> 1 THEN 11900 ELSE ZZ$=G$( 1 )
     : FIELD1, 80 *P% AS DX$, 80 AS DL$
     : LSET DL$=STRING$(255,250)
     : PUT 1,R%
     : GOSUB 26000
11890 GOSUB 1250
11900 REM ** BEGIN OUTPUT **
11995 FIELD 1, 80*P% AS TX$, 20AS F$(1), 20AS F$(2),
     20AS F$(3), 20AS F$(4)
11998 REM ** INSERT CHANGED FIELDS INTO RECORD
     AND SEND **
11999 FOR I=1 TO 4
     : LSET F$(I)=G$(I)
     : NEXT
     : PUT 1,R%
     : GOTO 60
12000 REM ** BEGIN THE RECORD DELETE ROUTINE **
12010 CLS
     : GOSUB 27000       'TRY TO FIND THE RECORD
12199 REM ** UNPACK FIELDS IN FILE FOR DISPLAY **
12200 GOSUB 28000
12799 REM ** DISPLAY RECORD IS MATCH ON KEY **
12800 GOSUB 29000
12900 REM ** DELETE CODE WRITTEN IN ALL FIELDS **
12905 R%=(RP-1)/ 3 +1
     : P%=(RP-1)-(R%-1)* 3
12910 FIELD 1, 80 *P% AS DX$, 80 AS DL$
     : LSET DL$=STRING$(255,250)
     : PUT 1,R%
     : GOTO 60
22001 DATA OBJECT
22002 DATA SERIAL #
22003 DATA LOCATION
22004 DATA COLOR
25000 REM ** BEGIN ERROR ROUTINE **
25001 IF ERL<10000 AND ERL>1000 THEN PRINT "PROBABLE
     ERROR IN EDIT SPECIFICATIONS."
25010 PRINT "ERROR ENCOUNTERED IN LINE";ERL
25020 PRINT "ERROR #=";ERR/2+1
     : CLOSE
     : RUN
25999 REM ** HASHING ALGORITHM SUBROUTINE **
26000 FOR ZZ=1 TO LEN(ZZ$)
26010 X#=X#+ZZ*ASC(MID$(ZZ$,ZZ,1))
26020 NEXT ZZ
26030 X#=X#*X#*X#*X#
     : X$=STR$(X#)
     : RP=VAL(MID$(X$,5,4))
     : X#=0
26040 RP= 1000 *RP/9999
     : RETURN
27000 REM ** LOOK FOR RECORD SUBROUTINE **
27010 GOSUB 1010
     : KF$=G$( 1 )
27020 OS=0
27030 ZZ$=KF$
     : GOSUB 26000  'GO TO HASHING ROUTINE, GET POS.
27040 RP=RP+1
     : IF RP> 1000 THEN RP=1 'NOT FOUND?
     BEGIN AGAIN @ #1
27045 R%=(RP-1)/ 3 +1
     : P%=(RP-1)-(R%-1)* 3
27046 FIELD 1, 80*P% AS TX$, 20AS F$(1),
     20AS F$(2), 20AS F$(3), 20AS F$(4)
27047 IF OS=1 THEN GET 1,R%
     : RETURN
27050 GET 1,R%
     : IF LEFT$(F$( 1 ),LEN(ZZ$))=ZZ$ THEN RETURN
27055 FIELD 1, 80 *P% AS DX$, 80 AS ZY$
     : GET 1,R%
     : IF ZY$=STRING$( 80 ,255) THEN PRINT "RECORD
     NOT FOUND."
     : RUN
27057 FIELD 1,80*P% AS TX$, 20AS F$(1),
     20AS F$(2), 20AS F$(3), 20AS F$(4)
27060 GOTO 27040  'NO MATCH
     : TRY NEXT RECORD
27999 REM ** UNPACK FIELDS IN RECORD **
28000 G$(1)=F$(1)
28010 G$(2)=F$(2)
28020 G$(3)=F$(3)
28030 G$(4)=F$(4)
28998 RETURN
28999 REM ** DISPLAY FOUND RECORD **
29000 CLS
     : IF OS=0 THEN PRINT "IS THIS IT?
     (PLEASE PRESS Y IF CORRECT)"
29010 FOR I=1 TO 4
29020 READ R$
     : PRINT R$;":"; G$(I); TAB(50);;"FIELD #";I
29030 NEXT
     : RESTORE
29035 IF OS=1 THEN RETURN
29050 AN$=INKEY$
     : IF AN$="" THEN 29050
29055 IF AN$<>"Y" THEN GOSUB 27040
     : GOSUB 28000
     : GOTO 29000
29999 REM ** NUMERIC FIELD EDIT CHECK SUBROUTINE **
30000 CD=INSTR(CD$, CHR$(32))
     : IF CD>1 THEN CD$=LEFT$(CD$,CD-1)
     +MID$(CD$,CD+1)
     : GOTO 30000
     : ELSE IF CD=1 THEN CD$=MID$(CD$,2)
     : GOTO 30000
30002 CD=INSTR(CD$,"-")
     : IF CD>0 AND INSTR(CD+1,CD$,"-")>0 THEN E=1
     : RETURN
30005 FOR ZZ=1 TO LEN(CD$)
30010 IF MID$(CD$,ZZ,1)<"0" OR MID$(CD$,ZZ,1)>"9"
     THEN IF MID$(CD$,ZZ,1)<>"."
     AND MID$(CD$,ZZ,1)<>"-" THEN E=1
     : RETURN
30020 NEXT ZZ
30030 RETURN
30999 REM ** ALPHA FIELD EDIT CHECK SUBROUTINE **
31000 FOR ZZ=1 TO LEN(CD$)
31010 IF (MID$(CD$,ZZ,1)<"A" OR MID$(CD$,ZZ,1)>"Z")
```

```
     AND MID$(CD$,ZZ,1)<>CHR$(32) THEN E=1
     : RETURN
31020 NEXT ZZ
31030 RETURN
32000 REM ** INITIALIZE THE HASHED FILE **
32010 PRINT "THIS WILL ERASE ALL PREVIOUS ENTRIES,
     IF ANY!"
32020 PRINT "TO CONTINUE INITIALIZATION,
     HIT THE C KEY"
32030 AN$=INKEY$
     : IF AN$="" THEN 32030 ELSE IF AN$<>"C" THEN RUN
32035 PRINT "THIS WILL TAKE A LITTLE WHILE."
32040 FIELD #1, 255 AS AZ$, 1 AS DZ$
     : LSET AZ$=STRING$(255,255)
     : LSET DZ$=CHR$(255)
32050 FOR I=1 TO 334
     : PUT 1,I
     : NEXT
     : RUN
35000 FOR K=0 TO 999
     : RP=K
     : OS=1
     : GOSUB 27040
35005 FIELD 1, 80*P% AS TX$, 20AS F$(1),
     20AS F$(2), 20AS F$(3), 20AS F$(4)
35010 IF F$( 1 )>CHR$(249) THEN 35990
35050 GOSUB 28000
     : REM UNPACK RECORD FOR DISPLAY
35960 GOSUB 29000
     : REM DISPLAY THE RECORD
35970 FOR J=1 TO 2000
     : NEXT J
     : REM WAIT A LITTLE BIT BEFORE NEXT RECORD
35990 NEXT K
     : GOTO 60
```

Records

```
1 REM ** PROGRAM NAME:RECORDS/BAS **
2 REM ** DATA FILE NAME:RECORDS/DAT:0 **
3 REM ** DATA FILE IS ON DRIVE 0 **
4 REM ** MAXIMUM FILE SIZE IS 50 RECORDS **
5 REM ** RECORD LENGTH IS 210 PACKED 1 PER SECTOR **
6 REM
7 REM ** BY THOMAS TREPANIER **
8 REM ** 35 LINCOLN AVE **
9 REM ** ARDSLEY, NY 10502 **
10 REM
11 REM CHANGE DISKS--REINITIALIZE HERE
20 CLEAR 3000
30 OPEN "R",1, "RECORDS/DAT:0"
40 ON ERROR GOTO 25001
50 DIMF$(22),G$(22)
60 FIELD 1,20AS F$(1),18AS F$(2),24AS F$(3),9AS
     F$(4),11AS F$(5),15AS F$(6),15AS F$(7),15AS
     F$(8),20AS F$(9),14AS F$(10),10AS F$(11),20AS
     F$(12),7AS F$(13),2AS F$(14),10AS F$(15)
70 FOR I=1 TO 22
     : G$(I)=""
     : NEXT
     : FC=0
     : UF=0
     : G$=""
     : CLS
80 PRINT TAB(18);"PRODUCED BY TOM TREPANIER"
85 PRINT
90 PRINT "                         OCCUPANCY RECORDS"
100 PRINT
110 PRINT "ENTER DATA.....................DEPRESS E"
120 PRINT "LOOK UP A RECORD...............DEPRESS L"
130 PRINT "SCAN ALL RECORDS ON FILE.......DEPRESS S"
140 PRINT "UPDATE A RECORD................DEPRESS U"
150 PRINT "DELETE A RECORD................DEPRESS D"
160 PRINT "INITIALIZE THE FILE............DEPRESS I"
170 PRINT "EXIT THE PROGRAM...............DEPRESS X"
180 PRINT "PLEASE DEPRESS THE LETTER OF YOUR CHOICE:
     NO ENTER KEY IS NEEDED"
190 PRINT "TO RETURN TO THE MENU FROM UPDATE OR ENTRY
     WITHOUT SAVING YOUR MATERIAL TO THE DISK, ENTER
     END FOR A FIELD ENTRY."
200 AN$=INKEY$
     : IF AN$="" THEN 200
210 IF AN$<>"E" AND AN$<>"L" AND AN$<>"U" AND
     AN$<>"D" AND AN$<>"I" AND AN$<>"X" AND AN$<>"S"
     THEN 60
220 IF AN$="E" THEN 1000
230 IF AN$="L" THEN 10000
240 IF AN$="S" THEN 35000
250 IF AN$="U" THEN 11000
260 IF AN$="D" THEN 12000
270 IF AN$="I" THEN 32000
280 CLOSE
     : NEW
1000 CLS
     : REM * BEGIN ENTRY *
1005 FC=FC+1
1006 ON FC GOSUB 1010, 1060, 1110, 1160, 1210, 1260,
     1310, 1360, 1410, 1460, 1510, 1560, 1610, 1660,
     1710
1007 IF FC< 15 THEN 1005 ELSE 1760
1010 PRINT "NAME:"
1020 LINE INPUTG$(1)
     : IF G$(1)="END" THEN 60
1030 IF LEN(G$(1))>20 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1010
1050 RETURN
1060 PRINT "ADDRESS:"
1070 LINE INPUT G$(2)
     : IF G$(2)="END" THEN 60
1080 IF LEN(G$(2))>18 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1060
1100 RETURN
1110 PRINT "CITY, STATE & ZIP CODE:"
1120 LINE INPUT G$(3)
     : IF G$(3)="END" THEN 60
1130 IF LEN(G$(3))>24 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1110
1150 RETURN
1160 PRINT "DATE OF BIRTH:"
1170 LINE INPUT G$(4)
     : IF G$(4)="END" THEN 60
1180 IF LEN(G$(4))>9 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1160
1200 RETURN
1210 PRINT "DOCTOR:"
1220 LINE INPUT G$(5)
     : IF G$(5)="END" THEN 60
1230 IF LEN(G$(5))>11 THEN PRINT "TOO LONG PLEASE
     ABREV."
```

```
     : GOTO 1210
1250 RETURN
1260 PRINT "SOCIAL SECURITY #:"
1270 LINE INPUT G$(6)
     : IF G$(6)="END" THEN 60
1280 IF LEN(G$(6))>15 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1260
1300 RETURN
1310 PRINT "MEDICARE:"
1320 LINE INPUT G$(7)
     : IF G$(7)="END" THEN 60
1330 IF LEN(G$(7))>15 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1310
1350 RETURN
1360 PRINT "MEDICAID:"
1370 LINE INPUT G$(8)
     : IF G$(8)="END" THEN 60
1380 IF LEN(G$(8))>15 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1360
1400 RETURN
1410 PRINT "NEXT OF KIN:"
1420 LINE INPUT G$(9)
     : IF G$(9)="END" THEN 60
1430 IF LEN(G$(9))>20 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1410
1450 RETURN
1460 PRINT "TELEPHONE:"
1470 LINE INPUT G$(10)
     : IF G$(10)="END" THEN 60
1480 IF LEN(G$(10))>14 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1460
1500 RETURN
1510 PRINT "RELIGION: "
1520 LINE INPUT G$(11)
     : IF G$(11)="END" THEN 60
1530 IF LEN(G$(11))>10 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1510
1550 RETURN
1560 PRINT "MEDICATION: "
1570 LINE INPUT G$(12)
     : IF G$(12)="END" THEN 60
1580 IF LEN(G$(12))>20 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1560
1600 RETURN
1610 PRINT "SEX: "
1620 LINE INPUT G$(13)
     : IF G$(13)="END" THEN 60
1630 IF LEN(G$(13))>7 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1610
1650 RETURN
1660 PRINT "ROOM #: "
1670 LINE INPUT G$(14)
     : IF G$(14)="END" THEN 60
1680 IF LEN(G$(14))>2 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1660
1700 RETURN
1710 PRINT "DIET: "
1720 LINE INPUT G$(15)
     : IF G$(15)="END" THEN 60
1730 IF LEN(G$(15))>10 THEN PRINT "TOO LONG PLEASE
     ABREV."
     : GOTO 1710
1750 RETURN
1760 ZZ$=G$(1)
     : GOSUB 26000
1770 REM ** LOOK FOR RECORD SPACE **
1780 GOSUB 1790
     : GOTO 1840
1790 RP=RP+1
     : IF RP> 50 THEN RP=1
1800 R%=(RP-1)/ 1 +1
     : P%=(RP-1)-(R%-1)* 1
1805 FIELD 1, 210 *P% AS DX$, 210 AS ZY$
1810 GET 1,R%
     : IF ZY$<CHR$(250) THEN 1790
1820 FIELD 1, 20AS F$(1), 18AS F$(2), 24AS F$(3),
     9AS F$(4), 11AS F$(5), 15AS F$(6), 15AS F$(7),
     15AS F$(8), 20AS F$(9), 14AS F$(10),
     10AS F$(11), 20AS F$(12), 7AS F$(13),
     2AS F$(14), 10AS F$(15)
1830 RETURN
1840 FOR K=1 TO 15
1850 LSET F$(K)=G$(K)
     : NEXT
     : PUT 1,R%
1860 PRINT "ARE YOU FINISHED(Y/N)"
1870 TM$=INKEY$
     : IF TM$="" THEN 1870 ELSE PRINT TM$
1880 REM ** IF DONE, END
     : IF NOT, RETURN TO MENU, ELSE REPEAT **
1890 IF TM$<>"Y" AND TM$<>"N" THEN PRINT "PLEASE
     ANSWER "; "Y"; " OR "; "N"
     : GOTO 1860
1900 IF TM$="Y" THEN 60
1910 FOR I=1 TO 22
     : G$(I)=""
     : NEXT
     : FC=0
     : UF=0
     : CLS
     : GOTO 1000
10000 REM ** BEGIN THE FILE LOOK-UP ROUTINE **
10010 CLS
     : GOSUB 27000      'TRY TO FIND THE RECORD
10199 REM ** UNPACK FIELDS IN FILE FOR DISPLAY **
10200 GOSUB 28000
10799 REM ** DISPLAY RECORD IF MATCH ON KEY **
10800 GOSUB 29000
10860 GOTO 60
11000 REM ** BEGIN THE FILE UPDATE ROUTINE **
11010 CLS
     : GOSUB 27000      'TRY TO FIND THE RECORD
11199 REM ** UNPACK FIELDS IN FILE FOR DISPLAY **
11200 GOSUB 28000
11799 REM ** DISPLAY RECORD IF MATCH ON KEY **
11800 GOSUB 29000
11810 PRINT "WHAT FIELD NUMBER DO YOU WANT TO
     UPDATE?"
11820 INPUT UF
11840 IF UF> 15 OR UF<1 THEN PRINT "INVALID FIELD"
     : GOTO 11810
11860 ON UF GOSUB 1010, 1060, 1110, 1160, 1210, 1260,
     1310, 1360, 1410, 1460, 1510, 1560, 1610, 1660,
     1710
11865 R%=(RP-1)/ 1 +1
     : P%=RP-1-(R%-1)* 1
11870 IF UF<> 1 THEN 11900 ELSE ZZ$=G$( 1 )
     : FIELD 1, 210 *P% AS DX$, 210 AS DL$
     : LSET DL$=STRING$(255,250)
     : PUT 1,R%
     : GOSUB 26000
11890 GOSUB 1790
11900 REM ** BEGIN OUTPUT **
11995 FIELD 1, 20AS F$(1), 18AS F$(2), 24AS F$(3),
     9AS F$(4), 11AS F$(5), 15AS F$(6), 15AS F$(7),
     15AS F$(8), 20AS F$(9), 14AS F$(10),
     10AS F$(11), 20AS F$(12), 7AS F$(13),
     2AS F$(14), 10AS F$(15)
11998 REM ** INSERT CHANGED FIELDS INTO RECORD
     AND SEND **
11999 FOR I=1 TO 15
```

```
     : LSET F$(I)=G$(I)
     : NEXT
     : PUT 1,R%
     : GOTO 6Ø
12ØØØ REM ** BEGIN THE RECORD DELETE ROUTINE **
12Ø1Ø CLS
     : GOSUB 27ØØØ       'TRY TO FIND THE RECORD
12199 REM ** UNPACK FIELDS IN FILE FOR DISPLAY **
122ØØ GOSUB 28ØØØ
12799 REM ** DISPLAY RECORD IS MATCH ON KEY **
128ØØ GOSUB 29ØØØ
129ØØ REM ** DELETE CODE WRITTEN IN ALL FIELDS **
12905 R%=(RP-1)/ 1 +1
     : P%=(RP-1)-(R%-1)* 1
1291Ø FIELD 1, 21Ø *P% AS DX$, 21Ø AS DL$
     : LSET DL$=STRING$(255,25Ø)
     : PUT 1,R%
     : GOTO 6Ø
22ØØ1 DATA NAME
22ØØ2 DATA ADDRESS
22ØØ3 DATA CITY STATE & ZIP
22ØØ4 DATA BIRTH DATE
22ØØ5 DATA DOCTOR
22ØØ6 DATA SOCIAL SECURITY
22ØØ7 DATA MEDICARE
22ØØ8 DATA MEDICAID
22ØØ9 DATA NEXT OF KIN
22Ø1Ø DATA TELEPHONE
22Ø11 DATA RELIGION
22Ø12 DATA MEDICATION
22Ø13 DATA SEX
22Ø14 DATA ROOM #
22Ø15 DATA DIET
25ØØØ REM ** BEGIN ERROR ROUTINE **
25ØØ1 IF ERL<1ØØØØ AND ERL>1ØØØ THEN PRINT "PROBABLE
     ERROR IN EDIT SPECIFICATIONS."
25Ø1Ø PRINT "ERROR ENCOUNTERED IN LINE";ERL
25Ø2Ø PRINT "ERROR #=";ERR/2+1
     : CLOSE
     : RUN
25999 REM ** HASHING ALGORITHM SUBROUTINE **
26ØØØ FOR ZZ=1 TO LEN(ZZ$)
26Ø1Ø X#=X#+ZZ*ASC(MID$(ZZ$,ZZ,1))
26Ø2Ø NEXT ZZ
26Ø3Ø X#=X#*X#*X#*X#
     : X$=STR$(X#)
     : RP=VAL(MID$(X$,5,4))
     : X#=Ø
26Ø4Ø RP= 5Ø *RP/9999
     : RETURN
27ØØØ REM ** LOOK FOR RECORD SUBROUTINE **
27Ø1Ø GOSUB 1Ø1Ø
     : KF$=G$( 1 )
27Ø2Ø OS=Ø
27Ø3Ø ZZ$=KF$
     : GOSUB 26ØØØ   'GO TO HASHING ROUTINE, GET POS.
27Ø4Ø RP=RP+1
     : IF RP>5Ø THEN RP=1 'NOT FOUND?
     BEGIN AGAIN @ #1
27Ø45 R%=(RP-1)/ 1 +1
     : P%=(RP-1)-(R%-1)* 1
27Ø46 FIELD 1, 2ØAS F$(1), 18AS F$(2), 24AS F$(3),
     9AS F$(4), 11AS F$(5), 15AS F$(6), 15AS F$(7),
     15AS F$(8), 2ØAS F$(9), 14AS F$(1Ø),
     1ØAS F$(11), 2ØAS F$(12), 7AS F$(13),
     2AS F$(14), 1ØAS F$(15)
27Ø47 IF OS=1 THEN GET 1,R%
     : RETURN
27Ø5Ø GET 1,R%
     : IF LEFT$(F$( 1 ),LEN(ZZ$))=ZZ$ THEN RETURN
27Ø55 FIELD 1, 21Ø *P% AS DX$, 21Ø AS ZY$
     : GET 1,R%
     : IF ZY$=STRING$( 21Ø ,255) THEN PRINT "RECORD
     NOT FOUND."
     : RUN
27Ø57 FIELD 1, 2ØAS F$(1), 18AS F$(2), 24AS F$(3),
     9AS F$(4), 11AS F$(5), 15AS F$(6), 15AS F$(7),
     15AS F$(8), 2ØAS F$(9), 14AS F$(1Ø),
     1ØAS F$(11), 2ØAS F$(12), 7AS F$(13),
     2AS F$(14), 1ØAS F$(15)
27Ø6Ø GOTO 27Ø4Ø  'NO MATCH
     : TRY NEXT RECORD
27999 REM ** UNPACK FIELDS IN RECORD **
28ØØØ G$(1)=F$(1)
28Ø1Ø G$(2)=F$(2)
28Ø2Ø G$(3)=F$(3)
28Ø3Ø G$(4)=F$(4)
28Ø4Ø G$(5)=F$(5)
28Ø5Ø G$(6)=F$(6)
28Ø6Ø G$(7)=F$(7)
28Ø7Ø G$(8)=F$(8)
28Ø8Ø G$(9)=F$(9)
28Ø9Ø G$(1Ø)=F$(1Ø)
281ØØ G$(11)=F$(11)
2811Ø G$(12)=F$(12)
2812Ø G$(13)=F$(13)
2813Ø G$(14)=F$(14)
2814Ø G$(15)=F$(15)
28998 RETURN
28999 REM ** DISPLAY FOUND RECORD **
29ØØØ CLS
     : IF OS=Ø THEN PRINT "IS THIS IT? (PLEASE PRESS
     Y IF CORRECT)"
29Ø1Ø FOR I=1 TO 15
29Ø2Ø READ R$
     : PRINT R$; ":"; G$(I); TAB(5Ø);;"FIELD #";I
29Ø3Ø NEXT
     : RESTORE
29Ø35 IF OS=1 THEN RETURN
29Ø5Ø AN$=INKEY$
     : IF AN$="" THEN 29Ø5Ø
29Ø55 IF AN$<>"Y" THEN GOSUB 27Ø4Ø
     : GOSUB 28ØØØ
     : GOTO 29ØØØ
29999 REM ** NUMERIC FIELD EDIT CHECK SUBROUTINE **
3ØØØØ CD=INSTR(CD$,CHR$(32))
     : IF CD>1 THEN CD$=LEFT$(CD$,CD-1)
     +MID$(CD$,CD+1)
     : GOTO 3ØØØØ
     : ELSE IF CD=1 THEN CD$=MID$(CD$,2)
     : GOTO 3ØØØØ
3ØØØ2 CD=INSTR(CD$,"-")
     : IF CD>Ø AND INSTR(CD+1,CD$,"-")>Ø THEN E=1
     : RETURN
3ØØØ5 FOR ZZ=1 TO LEN(CD$)
3ØØ1Ø IF MID$(CD$,ZZ,1)<"Ø" OR MID$(CD$,ZZ,1)>"9"
     THEN IF MID$(CD$,ZZ,1)<>"." AND
     DMID$(CD$,ZZ,1)<>"-" THEN E=1
     : RETURN
3ØØ2Ø NEXT ZZ
3ØØ3Ø RETURN
3Ø999 REM ** ALPHA FIELD EDIT CHECK SUBROUTINE **
31ØØØ FOR ZZ=1 TO LEN(CD$)
31Ø1Ø IF (MID$(CD$,ZZ,1)<"A" OR MID$(CD$,ZZ,1)>"Z")
     AND MID$(CD$,ZZ,1)<>CHR$(32) THEN E=1
     : RETURN
31Ø2Ø NEXT ZZ
31Ø3Ø RETURN
32ØØØ REM ** INITIALIZE THE HASHED FILE **
32Ø1Ø PRINT "THIS WILL ERASE ALL PREVIOUS ENTRIES,
     IF ANY!"
32Ø2Ø PRINT "TO CONTINUE INITIALIZATION,
     HIT THE C KEY"
32Ø3Ø AN$=INKEY$
     : IF AN$="" THEN 32Ø3Ø ELSE IF AN$<>"C" THEN RUN
32Ø35 PRINT "THIS WILL TAKE A LITTLE WHILE."
32Ø4Ø FIELD #1,255 AS AZ$,1 AS DZ$
     : LSET AZ$=STRING$(255,255)
     : LSET DZ$=CHR$(255)
32Ø5Ø FOR I=1 TO 5Ø
```

```
     : PUT 1,I
     : NEXT
     : RUN
35000 FOR K=0 TO 49
     : RP=K
     : OS=1
     : GOSUB 27040
35005 FIELD 1, 20AS F$(1), 18AS F$(2), 24AS F$(3),
     9AS F$(4), 11AS F$(5), 15AS F$(6), 15AS F$(7),
     15AS F$(8), 20AS F$(9), 14AS F$(10),
     10AS F$(11), 20AS F$(12), 7AS F$(13),
     2AS F$(14), 10AS F$(15)
35010 IF F$( 1 )>CHR$(249) THEN 35990
35050 GOSUB 28000
     : REM UNPACK RECORD FOR DISPLAY
35960 GOSUB 29000
     : REM DISPLAY THE RECORD
35970 FOR J=1 TO 2000
     : NEXT J
     : REM WAIT A LITTLE BIT BEFORE NEXT RECORD
35990 NEXT K
     : GOTO 60
```

Five Line Label

Sam Seltzer
2 Tudor Lane #2
Lockport, NY
14994

Recently I have read several programs for making address labels, using BASIC. I am submitting another such program because I believe it incorporates a few additional features which are most desirable—and the entire program length is only 34 lines.

This program allows the user to design an address label, up to 5 lines in length, preview it for editing before printing, and designate how many copies of the label are to be printed. It further allows the user to make additional copies, with corrections and/or changes if desired.

Lines 130-180 allow up to five lines of text to be entered.

Lines 190-210 display the original text of the label and provide an opportunity to change text.

Lines 230-270 allow for text changes; the use of ENTER will retain the original line text.

Line 280 permits re-editing the new text.

Line 290 establishes the number of copies of the label to be printed.

Lines 300-310 print the labels and keep count on the screen, indicating the number printed and the number yet to be printed.

Line 320 announces completion and asks if additional labels are desired.

Line 330 obtains particulars and branches accordingly.

```
10 'FIVE LINE LABEL MAKING PROGRAM
20 '
30 ' SAM SELTZER
40 ' 2 TUDOR LANE #2
50 ' LOCKPORT, NY
60 ' 14094
70 '
80 'THIS PROGRAM PRINTS A LABEL UP TO FIVE
     LINES IN LENGTH
90 '
100 CLS
     : CLEAR 1000
110 PRINT@275, "MAKE YOUR OWN LABELS"
     : PRINT
     : PRINT
120 FOR R=1 TO 1000
     : NEXT
130 PRINT "THIS IS A FIVE-LINE LABEL: BE SURE
     TO PRESS <ENTER> FOR A BLANK LINE"
     : PRINT
140 LINE INPUT "LINE 1                :   ";N$
     : IF LEN(N$)>30,140
150 LINE INPUT "LINE 2                :   ";E$
     : IF LEN(E$)>30,150
160 LINE INPUT "LINE 3                :   ";A$
     : IF LEN(A$)>30,160
170 LINE INPUT "LINE 4                :   ";X$
     : IF LEN(X$)>30,170
180 LINE INPUT "LINE 5                :   ";Z$
     : IF LEN(Z$)>30,180
190 CLS
     : PRINT "HERE IS HOW THE LABEL WILL LOOK:"
     : PRINT
     : PRINT
200 PRINT TAB(10)N$
     : PRINT TAB(10)E$
     : PRINT TAB(10)A$
     : PRINT TAB(10)X$
     : PRINT TAB(10)Z$
210 PRINT
     : PRINT "DO YOU WISH TO CHANGE ANYTHING";
     : INPUT C$
     : PRINT
220 IF C$="N" THEN GOTO 290 ELSE PRINT "WHICH DO
     YOU WISH TO CHANGE:"
230 LINE INPUT "LINE 1                :   ";NN$
     : IF NN$<>""THEN N$=NN$
240 LINE INPUT "LINE 2                :   ";EE$
     : IF EE$<>""THEN E$=EE$
250 LINE INPUT "LINE 3                :   ";AA$
     : IF AA$<>""THEN A$=AA$
260 LINE INPUT "LINE 4                :   ";XX$
     : IF XX$<>""THEN X$=XX$
270 LINE INPUT "LINE 5                :   ";ZZ$
     : IF ZZ$<>""THEN Z$=ZZ$
280 GOTO 190
290 PRINT
     : PRINT "HOW MANY LABELS DO YOU WANT";
     : INPUT N
300 T=0
     : FOR X=1 TO N
     : LPRINT CHR$(27)"E"N$
     : LPRINT E$
     : LPRINT A$
     : LPRINT X$
     : LPRINT Z$
     : T=T+1
     : LPRINT
     : PRINT@900, USING"##";T;
     : PRINT "  DONE - ";N-T;"TO GO  "
310 NEXT
320 PRINT "THOSE ARE DONE. DO YOU WANT ANY MORE";
     : INPUT Q$
330 IF Q$="Y" CLS
     : INPUT "THE SAME LABEL";QQ$
     : IF QQ$="Y" THEN 190 ELSE CLS
     : GOTO 140
340 CLS
     : END
```

AutoDial Program for Modem II

Mark L. Sills
539 Elvis Drive
San Jose, CA 95123

I am enclosing a copy of my version of the AutoDial Program for the Modem II. Once the user has set the system clock and invoked SETCOM this BASIC program enables the user to select from a menu of frequently called services. The modem is programmed and the user is able to further choose to either call at that moment or at a later time. When the system time equals the desired calling time, the modem will be instructed to dial.

(Editor's Note: If you have problems running the program, check to make sure that your Modem II is in the programmable mode. If the modem is in the programmable mode, only the ON and TR lights will be on. See your Modem II manual for further information.)

```
1 '* * * * * * * * * * * * * * * * * * * * * * * * * *
2 '* F/S=SMTMODEM/BAS AUTODIAL PROGRAM FOR MODEM II*
3 '* VERSION 5.Ø LAST REVISION FEBRUARY 19, 1983 * *
4 '* REFER TO TRS-8Ø MICROCOMPUTER NEWS DEC. 1982  *
5 '* MODIFIED BY MARK L. SILLS                     *
6 '* * * * * * * * * * * * * * * * * * * * * * * * * *
7 '* * * INITIALIZE  * * *
8 CLS
    : CLEAR 1ØØ
9 DIM I!,IK$,SN%,EC%,L!,PG$,TR!,A!,DT$,PN$,ED$,
    H$,T$,TS$
1Ø ' * * CREATE SPECIAL CHARACTER MODEL III  * *
11 POKE 16420,1
12 H$ = CHR$(244) + CHR$(245) + CHR$(246) + CHR$(32)
13 ' * * GET TELEPHONE NUMBER DATE * *
14 PRINT@1Ø, "AUTODIAL SYSTEM        SYSTEM TIME   ";
    : PRINT H$
15 ' * INSERT DESIRED NAMES IN PRINT STATEMENTS *
16 PRINT@133,"1)"
17 PRINT@153,"2)"
18 PRINT@197,"3)"
19 PRINT@217,"4)"
2Ø PRINT@261,"5)"
21 PRINT@281,"6)"
22 PRINT@325,"7)"
23 PRINT@345,"8) MANUAL DIAL"
24 PRINT
    : PRINT
25 PRINT "Please Make Your Choice :";
    : GOSUB 72
    : A = VAL(IK$)
    : PRINT A
26 ON A GOTO 65,66,67,68,69,7Ø,71,72
27 LINE INPUT "Enter Telephone Number :   ";PN$
28 PG$ = "D"+"T"+PN$
29 ' * ENTER PROGRAM MODE BY SENDING AN "*" *
3Ø TR = Ø
31 SN% = CINT(ASC("*"))
32 OUT &HEB,SN%
33 FOR I = 1 TO 5Ø
    : NEXT I
34 EC% = INP(&HEB)
35 TR = TR+1
36 IF EC% < > SN% AND TR <3 THEN 31
37 IF TR = 3 THEN CLS
    : PRINT "Can Not Program the Modem!!"
    : STOP
38 ' * SEND PHONE NUMBER DATA STRING TO MODEM II *
39 FOR L = 1 TO LEN (PG$)
4Ø SN% = CINT(ASC(MID$(PG$,L,1)))
41 OUT &HEB,SN%
42 FOR I = 1 TO 5Ø
    : NEXT I
43 EC% = INP(&HEB)
44 IF SN% < > EC% THEN PRINT"Can Not Program Modem!!"
    : STOP
45 NEXT L
46 ' * GET THE TIME FOR THE NUMBER TO BE DIALED *
47 ' * SEND "X" TO MODEM AT CORRECT TIME *
48 LINE INPUT "Modem Programmed, Enter the Time
    to dial (HH:MM) ";DT$
    : IF LEN (DT$) = Ø THEN 5Ø
49 TS$ = LEFT$ (RIGHT$ (TIME$ ,8),5)
    : IF DT$ < > TS$ THEN 49
5Ø SN% = CINT (ASC("X"))
51 OUT &HEB,SN%
52 FOR I = 1 TO 5Ø
    : NEXT I
53 ED% = INP (&HEB)
54 ' * TIMING LOOP FOR MODEM II PROTOCOL *
55 FOR I = 1 TO 3ØØØ
    : NEXT I
56 ' * TEST FOR CARRIER DETECT *
57 FOR I = 1 TO 25ØØ
    : IF (INP (&HØE8) AND &H2Ø) = Ø THEN 59
    : ELSE NEXT I
58 GOTO 8
59 CLS
    : PRINT "Connection Made !"
    : FOR I = 1 TO 3ØØ
    : NEXT I
62 FOR I = 1 TO 5
    : LPRINT CHR$(7)
    : NEXT I
    : CMD"S = MODEM"
63 CMD "S = MODEM"
64 ' * INSERT DESIRED PHONE NUMBERS BETWEEN QUOTES *
65 PN$ = ""
    : GOTO 28
66 PN$ = ""
    : GOTO 28
67 PN$ = ""
    : GOTO 28
68 PN$ = ""
    : GOTO 28
69 PN$ = ""
    : GOTO 28
7Ø PN$ = ""
    : GOTO 28
71 PN$ = ""
    : GOTO 28
72 IK$ = INKEY$
    : IF IK$ = "" THEN 72
    : ELSE RETURN
73 END
```

Peripherals

Radio Shack continues to increase the number of peripherals available for the TRS-80 computer range, to the extent that we all have difficulties deciding "what goes with what" on occasions. Whether you are considering adding more peripherals to your system or trying to get the best out of what you already have, we hope that the following brief points will help you with problems you have come across or help you avoid problems in the future.

PRINTERS

A printer is probably the most important peripheral you have purchased, or is the next one on your shopping list. A few general points first.

INTERFACES

Your computer "talks" to a printer through an interface, which can be either serial or parallel in nature.

If you have a Color Computer, then a serial interface built into the computer will allow you to talk to a printer which has a similar interface also built into it. The smaller DMP series printers of course have just such an interface specially designed for use with the CoCos. BUT do not assume that every computer with a serial interface will drive the DMPs serially, or that a CoCo will drive anyone else's serial printer. You might be lucky, but on the other hand you might not.

Most of our computers have parallel interfaces and of course will drive our parallel printers. Parallel interfacing with non-Radio Shack products may well work like a charm, but if it doesn't, it must be understood that the help that Computer Customer Service can give you, is going to be of a general rather than a specific nature.

CHOOSING PRINTERS

The choice is a trifle bewildering at times, so if you are about to buy one of our printers, then the following may help you arrive at a good decision. You may have need for several distinct features in your printer, such as letter quality, graphics, color, or fast report writing. Whereas two printers and a printer interface switch is one answer, a printer which does several things reasonably well can be a more economical solution.

The DMP-2100 combines nearly perfect letter quality print as a software selectable option and lower quality fast printing as another. In a situation which allows software spooling, the 2100 is almost unbeatable—it can switch from one to the other without physical intervention. If you choose the option of two printers, say a daisy wheel for correspondence and a dot-matrix printer for reports, and you are spooling, physically making the switch (for example on a printer interface switch) at the right time can be a problem.

The DWP-210 by comparison can be used to great effect as a combination printer at much lower cost, for lighter duty work. A bi-directional tractor can be added for report printing, and although you cannot speed up the printer, spooling can free up your computer for more throughput. If your software does not support spooling then our hardware spooler (PTC-64) may be the answer. Be aware though that special modes such as graphics or proportional space printing can give strange results on all printers unless you have set the PTC-64 to IGNORE mode. In the IGNORE mode the PTC-64 passes all codes through to the printer without interpretation.

The LMP-2150, by contrast, is a very heavy duty printer. It is designed for continuous report printing work, fewer ribbon changes, etc., and a long long life, and yet still have the desirable options for graphics and correspondence. An industry standard for comparison of one printer to another rates this Line Matrix Printer as capable of printing at 150 lines per minute (LPM), in data processing mode. This is nearly twice as fast as any previous Radio Shack printer. This is so fast that there is going to be a much greater chance of the printer "sitting around" waiting on the software to give it more work to do. Even the LMP-2150 cannot print at 150 LPM if the software driving it is running at 100 LPM. As with the DWP-210, software or hardware spooling can maximize the effectiveness of the 2150.

The DMP-420 is yet another good combination printer, for correspondence mode, data processing, and graphics, and adds another mode which is a new feature. This is a mode in which each character received by the printer is printed in hexadecimal representation. If you have ever written a graphics program or a printer driver, and you wanted to see what the printer got (as opposed to what you thought you sent), then this feature is for you.

And for color, well we simply cannot pass over the CGP-220 ink-jet printer. It is a perfect addition to a Tandy 2000 color system, or a Color Computer system, giving the ability to represent your color graphics screen on paper. Even if your TRS-80 doesn't have a color graphics screen option, you can (through your own software) still obtain seven-color graphics printouts. In the text mode of the CGP-220, and with the interchangeable black ink-pack, your can still get excellent program listings, or printed reports. While the text mode is not fast (37 characters per second), the printer is almost unbelievably quiet.

This article could go into detail about all our other printers, but suffice it to say that they all have their particular mix of features, constraints and so on. You need to balance your needs against price and throughput of the printer, keeping in mind the computer and software you will be using. A copy of the current Radio Shack Computer Catalog (RSC-11), available at Radio Shack Stores, Computer Centers, or participating dealers, will help you distinguish between the features of the various printers.

BAR-CODE READER

The bar-code reader is a peripheral which can integrate the Model 100 computer into many more projects than the phrase "Micro Executive Workstation" implies. Inventory, billing, pricing, and a long list of other tasks, are made more convenient, and can be speeded up immensely. We describe this as an easy-to-use device and it really is. However it will take a little practice and co-ordination before you can read a code first-time every time. One of the secrets is finding just the right speed for the wand to travel at, and a second secret is to find the appropriate angle between the wand and the paper.

The software driver, which Radio Shack provides with the wand, only really does the reading. If you wish to integrate what you have read into your data processing system, you will need to have your own programs to do it. Your program would of course call one of the drivers that can do the reading for you.

If the bar code reader is not what you need then maybe the CR-510 CARD READER is. The CR-510 is very intelligent. Through software it can be configured to match a wide range of RS232 interface options as well as a large array of card formats. The card reader is certainly good for grading multiple choice test scores, but can also be very useful in inventory, cataloging, or dozens of other situations. The card reader does have an advantage in a situation where lots of people have only a few cards each to mark or where prepared cards can be re-used very often.

COMMUNICATIONS

Not all that much new has come out lately in the way of new communications peripherals, so here is a new thought about an old one.

The DC-1200 MODEM when transferring TRS-XENIX object files may hang up and drop the line. The reason is that "C" compiler has a predilection for (apparently) padding files with bunches of control D characters. Now none of us had forgotten that the DC-1200 is supposed to hang up after three control Ds in succession, had we? The solution of course is to use a different modem to transfer object files, or to transmit the source code and have it compiled at the other end.

MISCELLANY

DMP series printers can give Printer Not Ready errors on the II/12/16 family, when an older LP series or slower printer may not do so. The solution is almost always to be found by DOing LPII, or LP2, or by operating TRSDOS 4.2 with FORMS "O" on, whichever is appropriate to the operating system being run. Also note that the DO LPII, or LP2, solution will not have any effect until the system is reset, and a new copy of the resident operating system is loaded into memory.

The CR-510 errors out frequently after handling a moderate number of mark sense cards correctly. The problem is probably that the rubber roller is being coated with graphite (pencil lead). The solution is to clean the roller frequently to prevent the build up of graphite. (see pages 15 & 16 of the CR-510 owners manual).

The Model 100 refuses to talk to a Mod 12 or 16B when hard wired but talks just fine over modems. Slight RS232C protocol differences mean that the Model 100 needs to have minor adjustments made at a Repair Center.

The DMP-2100 Sheet Feeder does not seem to recognize page length properly when running from SCRIPSIT 2.1 on the Model II/12. The solution is a SCRIPSIT patch found in Appendix A of the sheet feeder Operation Manual.

PATCH SCRIPSIT/SYS R=175 B=22 F=0F C=FF

This patch does not apply to the SCRIPSIT HD or Thinline versions.

This list of hints could go on almost endlessly but enough is enough for this time anyway.

Computer Customer Service Address and Phone Numbers

8AM to 5PM Central Time
Computer Customer Services
400 Atrium, One Tandy Center
Fort Worth, Texas 76102

Productivity/Special Applications	(817)338-2390
Accounting Software	(817)338-2391
O/S and Languages, Group No. 1	(817)338-2392
O/S and Languages, Group No. 2	(817)338-2393
Hardware and Communications	(817)338-2394
Home Software	(817)338-2395
Educational Software	(817)338-2396
Newsletter Subscription Problems	(817)870-0407

Line Printer Set-Up

William E. Allen
2912 Metairie Ct., Apt. 24
Metairie, LA 70002

Since the original LPSU program was published (original LPSU published Nov. 1982), it has undergone several revisions for use with my LP-IV and has received some additions for my CGP-115.

Some people think the CGP-115 is for graphics work only. Guess again. I find it equally adaptable for listings of assembly language programs. I formerly used my QP-II for this, but that aluminized paper just does not offer the convenience of 4" wide "conventional" paper for documentation purposes.

```
1 ' Line Printer Set-Up Program V. 1.5 (Title - LPSU)
2 '
3 '     By:     William E. Allen
4 '             2912 Metairie Ct. # 24
5 '             Metairie, LA  70002
6 '
10 CLEAR
    : ' To make room for input variables
15 POKE 16424,66
    : ' Sets Model I for 66 lines per page
16 ' For Model III, poke 67
20 CLS
25 PRINT
    : PRINT "Letter inputs may be either UPPER
    or lower case."
30 PRINT
    : PRINT "LINE PRINTER SET UP COMMANDS  Type #"
40 PRINT "FOR QUICK PRINTER II                 2"
50 PRINT "FOR LINE PRINTER IV                  4"
```

```
60 PRINT "FOR LINE PRINTER VII                    7"
61 PRINT "For Color Graphic Printer - 115       115"
70 W=0
     : PRINT
     : INPUT "<ENTER> WHICH PRINTER "; W
     : CLS
80 IF W=2 THEN 200
90 IF W=4 THEN 300
100 IF W=7 THEN 500
110 IF W=115 THEN 800
190 GOTO 20
200 CLS
     : PRINT "FOR QUICK PRINTER II"
210 PRINT "CARRIAGE RETURN    1"
220 PRINT "DOUBLE WIDTH       2"
230 C=0
     : INPUT "<ENTER> CHOICE ";C
240 IF C=1 THEN GOSUB 1180
     : END
250 IF C=2 THEN GOSUB 1140
     : END
260 GOTO 200
300 CLS
     : PRINT "FOR LINE PRINTER IV      Type #"
310 PRINT "DOUBLE WIDTH                1"
320 PRINT "PROPORTIONAL                2"
330 PRINT "STANDARD PRINT              3"
340 PRINT "CONDENSED PRINT             4"
350 PRINT "START UNDERLINE             5"
360 PRINT "STOP UNDERLINE              6"
365 PRINT "Combination printing        9"
370 C=0
     : PRINT
     : INPUT "<ENTER> CHOICE ";C
380 IF C=1 THEN CLS
     : PRINT CHR$(23)
     : GOSUB 1000
     : GOTO 9000
390 IF C=2 THEN GOSUB 1020
     : END
400 IF C=3 THEN GOSUB 1040
     : END
410 IF C=4 THEN GOSUB 1060
     : END
420 IF C=5 THEN GOSUB 1140
     : END
430 IF C=6 THEN GOSUB 1160
     : END
440 IF C=9 THEN CLS
     : GOTO 600
450 GOTO 300
500 CLS
     : PRINT "FOR LINE PRINTER VII         Type #"
510 PRINT "CARRIAGE RETURN/LINE FEED        1"
520 PRINT "GRAPHIC MODE                     2"
530 PRINT "STANDARD PRINT                   3"
540 PRINT "DOUBLE WIDTH                     4"
550 C=0
     : PRINT
     : INPUT "<ENTER> CHOICE "; C
560 IF C=1 THEN GOSUB 1160
     : END
570 IF C=2 THEN GOSUB 1120
     : END
580 IF C=3 THEN GOSUB 1100
     : END
590 IF C=4 THEN GOSUB 1080
     : END
591 GOTO 500
600 PRINT "Available combinations:        Type #"
610 PRINT "Double Width/Proportional        1"
620 PRINT "Double Width/Condensed           2"
630 PRINT "Anything WITH Underline          3"
640 PRINT "Anything, STOP Underline         4"
650 C=0
     : PRINT
     : INPUT "<ENTER> CHOICE ";C
660 IF C=1 THEN CLS
     : PRINT CHR$(23)
     : GOSUB 1020
     : GOSUB 1000
     : GOTO 9000
670 IF C=2 THEN CLS
     : PRINT CHR$(23)
     : GOSUB 1060
     : GOSUB 1000
     : GOTO 9000
680 IF C=3 THEN GOSUB 1140
     : PRINT
     : GOTO 310
690 IF C=4 THEN GOSUB 1160
     : PRINT
     : GOTO 310
691 CLS
     : GOTO 600
700 CLS
     : PRINT "FOR LINE PRINTER VII          Type #"
710 PRINT "CARRIAGE RETURN/LINE FEED         1"
720 PRINT "GRAPHIC MODE                      2"
730 PRINT "STANDARD PRINT                    3"
740 PRINT "DOUBLE WIDTH                      4"
750 C=0
     : PRINT
     : INPUT "<ENTER> CHOICE "; C
760 IF C=1 THEN GOSUB 1160
     : GOTO 9000
770 IF C=2 THEN GOSUB 1120
     : GOTO 9000
780 IF C=3 THEN GOSUB 1100
     : GOTO 9000
790 IF C=4 THEN GOSUB 1080
     : GOTO 9000
791 GOTO 700
800 CLS
     : PRINT
     : PRINT "For Color Graphic Printer 115   Type #"
810 PRINT "Text mode (to confirm mode)              1"
820 PRINT "Text mode (from Graphic mode)            2"
830 PRINT "Graphic mode (from Text mode)            3"
840 PRINT "Change pen color (from Graphic mode)     4"
850 PRINT "Change pen color (from Text mode)        5"
900 C=0
     : PRINT
     : INPUT "<ENTER> choice "; C
910 IF C=1 THEN GOSUB 2000
     : GOTO 9000
920 IF C=2 THEN GOSUB 2040
     : GOTO 9000
930 IF C=3 THEN GOSUB 1120
     : GOTO 9000
940 IF C=4 THEN GOSUB 2060
     : GOTO 9000
950 IF C=5 THEN GOSUB 2020
     : GOTO 9000
960 GOTO 800
1000 LINEINPUT "TYPE MATERIAL TO BE PRINTED
     DOUBLE WIDTH (MAX 40 CHARACTERS)"; A$
     : LPRINT CHR$(27); CHR$(14) A$
1001 ' 40 Characters = effective line limit for
     paper, not the program.
1010 RETURN
1020 LPRINT CHR$(27); CHR$(17)
1030 RETURN
1040 LPRINT CHR$(27); CHR$(19)
1050 RETURN
1060 LPRINT CHR$(27); CHR$(20)
1070 RETURN
1080 LPRINT CHR$(31)
1090 RETURN
1100 LPRINT CHR$(30)
1110 RETURN
1120 LPRINT CHR$(18)
```

```
113Ø RETURN
114Ø LPRINT CHR$(15)
115Ø RETURN
116Ø LPRINT CHR$(14)
117Ø RETURN
118Ø LPRINT CHR$(13)
119Ø RETURN
2ØØØ ' Begin special CGP-115 Section
2ØØ1 LPRINT CHR$(17)
2Ø1Ø RETURN
2Ø2Ø LPRINT CHR$(29)
2Ø3Ø RETURN
2Ø4Ø LPRINT "A"
2Ø5Ø RETURN
2Ø6Ø CLS
     : PRINT
     : PRINT "Type the number for the
     corresponding color."
     : GOSUB 2Ø61
     : PRINT
     : P=4
     : INPUTP
     : GOTO 2Ø66
2Ø61 PRINT
     : ' P = Pen #
2Ø62 PRINT "Black    Ø"
2Ø63 PRINT "Blue     1"
2Ø64 PRINT "Green    2"
2Ø65 PRINT "Red      3"
     : RETURN
2Ø66 IF P = Ø THEN LPRINT "CØ"
     : RETURN
2Ø67 IF P = 1 THEN LPRINT "C1"
     : RETURN
2Ø68 IF P = 2 THEN LPRINT "C2"
     : RETURN
2Ø69 IF P = 3 THEN LPRINT "C3"
     : RETURN
2Ø7Ø GOTO 2Ø6Ø
9ØØØ CLS
9Ø1Ø PRINT @ 4ØØ," "
9Ø2Ø C$=""
     : INPUT "Is Line Printer Set-Up Finished <Y/N>";
     C$
9Ø3Ø IF C$="Y" OR C$="y" THEN 91ØØ
9Ø4Ø IF C$="N" OR C$="n" THEN 2Ø
9Ø5Ø IF C$<>"Y" OR C$<>"y" OR C$<>"N" OR C$<>"n" THEN
     9ØØØ
91ØØ E$=""
     : INPUT "Stay in Basic <B> or return to DOS
     <D>"; E$
911Ø IF E$="B" OR E$="b" THEN NEW
912Ø IF E$="D" OR E$="d" THEN CMD"S"
913Ø IF E$<>"B" OR E$<>"b" OR E$<>"D" OR E$<>"d" THEN
     CLS
     : PRINT @ 4ØØ," "
     : GOTO 91ØØ
```

Model 100 Printer Formatter

Dave Cloutier
Bullard Road
North Brookfield, MA 01535
CompuServe ID: 70126,1515

The Model 100 has an excellent word processor built into its ROM. However, it does have rather primitive printing features, allowing only width control and no control characters.

The following program (about 1861 bytes in length) imitates a few of the basic features of SCRIPSIT-margin control, page length, and line spacing. The codes are displayed below.

Left margin	LM =	5 (default value)
Right margin	RM =	72
Top margin	TM =	16
Bottom margin	BM =	160
Page length	PL =	166
Line spacing	LS =	11

To use the control codes, type them after a greater-than sign (">"). They can be anywhere in a file, but they must be in upper case, separated from each other by one space apiece, and they must be on a separate line.

Example:

>LM=10 RM=40 LS=2 PL=30
Now is the time for all good foxes
to jump over lazy dogs.
>LS=1
TRS-80 computers are great.

The left margin is set to 10, and the right to 40, giving a line length of 30. The page length is set to 30 lines, but the top and bottom margins remain the default values of 6 and 60. Also, the file is double spaced. This is changed back to single spacing before the printing of the last line.

This program is a cheap, but useful alternative from the more advanced text formatters which are beginning to be introduced.

```
1Ø CLS
     : MAXFILES=1
     : GOSUB 1ØØØ
     : p$=chr$(27)+"p"
     : q$=chr$(27)+"q"
2Ø ?p$"WORD...."q$" by Dave Cloutier..."
     : FILES
     : INPUT "File to print";F$
     : IF F$="" THEN 2Ø
3Ø OPEN F$ FOR INPUT AS1
4Ø ON ERROR GOTO 99Ø
5Ø GOSUB 17ØØ
7Ø GOSUB 15ØØ
     : IF LEFT$(L$,1)=">" THEN GOSUB 155Ø
1ØØ LP=-1
     : IF TM<>Ø THEN FOR X=1 TO TM
     : LPRINT CHR$(1Ø);
     : LP=LP+1
     : NEXT
11Ø IF L$<>"><" THEN GOSUB 165Ø
     : LPRINT TAB(LM);L$
     : LP=LP+1
     : IF LS<>1 THEN FOR C=1 TO LS-1
     : LPRINT CHR$(1Ø);
     : LP=LP+1
     : IF LP=PL THEN 5ØØ ELSE NEXT C
12Ø FOR P=TM TO BM
13Ø GOSUB 15ØØ
     : IF LEFT$(L$,1)=">" THEN P=P-1
     : GOSUB 155Ø
15Ø IF L$<>"><" THEN GOSUB 165Ø
     : LPRINT TAB(LM);L$
     : LP=LP+1
     : IF LS<>1 THEN FOR C=1 TO LS-1
     : LPRINT CHR$(1Ø);
     : LP=LP+1
     : IF LP=PL THEN 5ØØ ELSE NEXT C
17Ø IF LP=>BM THEN IF LP<>PL-1 THEN FOR C=LP TO PL-2
     : LPRINT CHR$(1Ø);
     : NEXT C
     : GOTO 5ØØ ELSE GOTO 5ØØ
```

```
180 NEXT P
    : IF LP<>PL-1 THEN FOR X=LP TO PL-2
    : LPRINT CHR$(10);
    : NEXT X
500 BEEP
    : INPUT "Press <ENTER> for next page";Q$
510 GOTO 70
550 BEEP
    : BEEP
    : LPRINT TAB(LM);R$
    : IF LP<>PL-1 THEN FOR X=LP TO PL-2
    : LPRINT CHR$(10);
    : NEXT X
560 PRINT"End of file..."
    : END
    : MENU
990 IF ERL>1549 AND ERL<1631 THEN BEEP
    : PRINT "Syntax error in line settings" ELSE
    PRINT "Error";ERR;"in line";ERL
995 END
1000 LM=5
    : RM=72
    : TM=6
    : BM=60
    : PL=66
1010 LS=1
    : LN=67
1020 RETURN
1500 L$=R$
    : R$=""
    : X=LEN(L$)
1510 IF EOF(1) THEN 550
1520 A$=INPUT$(1,1)
    : X=X+1
    : IF A$=CHR$(10) THEN RETURN
1525 IF A$=CHR$(13) THEN A$=" "
1530 L$=L$+A$
    : IF X=LN THEN RETURN
    :  ELSE GOTO 1510
1550 IF LEN(L$)>1 THEN L$=RIGHT$(L$,LEN(L$)-1)+".."
1560 IF LEFT$(L$,1)<>"." THEN C$=LEFT$(L$,2)
     ELSE L$="><"
    : RETURN
1570 M$=MID$(L$,4,1)
    : IF ASC(M$)>47 AND ASC(M$)<58 THEN N$=M$
    : N=VAL(N$)
1580 IF MID$(L$,5,1)=" " OR MID$(L$,5,1)="." THEN
    L$=RIGHT$(L$,LEN(L$)-5)
    : GOTO 1600
1590 M$=MID$(L$,5,1)
    : N$=N$+M$
    : N=VAL(N$)
    : L$=RIGHT$(L$,LEN(L$)-6)
1600 IF C$="LM" THEN LM=N ELSE IF C$="RM" THEN RM=N
1610 IF C$="LS" THEN LS=N ELSE IF C$="PL" THEN PL=N
1620 IF C$="TM" THEN TM=N ELSE IF C$="BM" THEN BM=N
1630 LN=RM-LM
    : GOTO 1560
1650 R$=L$
    : IF LEN(L$)=0 THEN RETURN
1660 IF RIGHT$(L$,1)<>" " THEN L$=LEFT$(L$,LEN(L$)-1)
    : IF LEN(L$)<>0 THEN 1660 ELSE L$=R$
1670 R$=RIGHT$(R$,LEN(R$)-LEN(L$))
1680 RETURN
1700 INPUT "Test for errors";Q$
1710 IF LEFT$(Q$,1)<>"Y" AND LEFT$(Q$,1)<>"y"
     THEN PRINT "No test made."
    : RETURN
1720 PRINT "Testing for errors..."
    : B$=CHR$(10)
1730 IF NOT EOF(1) THEN A$=INPUT$(1,1) ELSE GOTO 1780
1750 IF A$=">" AND B$=CHR$(10) THEN LINEINPUT#1,L$
    : L$=">"+L$
    : GOSUB 1550
    : IF LM=>RM THEN PRINT "Error in left or right
    margin"
    : GOTO 1795
1755 B$=A$
    : IF BM>PL THEN PRINT "Error in bottom margin or
    page length"
    : GOTO 1795
1760 IF TM=>BM THEN PRINT "Error in top or bottom
    margin"
    : GOTO 1795
1765 IF LS>PL THEN PRINT "Line spacing too large"
    : GOTO 1795
1770 GOTO 1730
1780 PRINT "Testing completed"
    : GOSUB 1000
1790 CLOSE1
    : OPEN F$ FOR INPUT AS1
    : RETURN
1795 INPUT "Continue";Q$
    : Q$=LEFT$(Q$,1)
    : IF Q$="Y" OR Q$="y" THEN 1730 ELSE END
```

PC-2 Word Processor

John M. Dlugosz
2629 Glenhaven Road
Plano, TX 75023

Word Processor is a fairly simple program and is a compromise between memory for text storage and editing features.

After you have entered the Word Processor program (and saved it to tape we trust), you are ready to begin using it. Put the PC-2 in the RUN mode and type RUN. The display will show BUSY in the upper left corner. Begin typing your entry, being careful not to hold the (ENTER) key too long. If you hold the (ENTER) key down too long the program will load blank lines into memory. To make sure this has not happened, press the up arrow key until a "1" appears at the right of the display.

All letters are unshifted lowercase. You can use the shift key to enter uppercase letters; however, do not shift for the punctuation marks. The shift key and the punctuation marks will result in "/", "*", "–", and "+".

When your line entry exceeds 18 characters, the program will automatically break the entry at the last space and put the rest of the entry on the next line. If you want to end a line before reaching the 18 character limit, press the (ENTER) key at the point where you want the break. Pressing the (ENTER) key will force a new line. The line counter will disappear but you can restore it by pressing the up arrow. Because text is held in a buffer until the program advances to the next line, you should not use the down arrow to force a new line. Forcing a new line using the down arrow will result in lost text as the next line is loaded into the buffer. The arrows can be used to return to previous lines for editing purposes but the corrections will not be remembered until the (ENTER) key is pressed.

Once you have completed your entry, press the (MODE) key for a printout. If you want to add to or edit the text, use (DEF) A. (A lower case "a" will appear in the display and must be deleted, using the (DEL) key, before beginning new entry.) To edit the text, use the up arrow to return to the desired line. During the edit process extra cursors may be left behind. This will not affect the text and will not take up a space. Pressing the (DEL) key after using the up

or down arrow and before pressing any other key should eliminate these extra spaces.

If you want to save your text to tape, press DEF A then SHIFT MODE. You will be prompted for a file name then asked whether you want to Save "S" or Load "L". Type S to save the text. Use the same procedure to Load saved text except respond with L for Load.

```
10 CLEAR
     : DIM W$(50)*18,H$(0)*30
15 "A"WAIT 0
     : ON ERROR GOTO 20
20 A$=INKEY$
     : IF A$=""THEN 20
21 IF ASC A$=1 LET A=1
     : GOTO 20
22 IF A=1 GOTO 24
23 IF A$>="A" AND A$<="Z" LET A$=CHR$ (ASC A$+32)
     : GOTO 34
24 B=ASC A$
     : IF B=13 LET W$(P)=H$(0),P=P+1,H$(0)="",X=0,C=0
     : CLS
     : GOTO 20
25 IF B=8 LET X=X-1,H$(0)=LEFT$ (H$(0),X),C=X-1
     : CLS
     : PRINT H$(0);CHR$ 127
     : GOTO 20
26 IF B>16 AND B<23 LET A$=CHR$ (B+16)
27 IF B>41 AND B<48 AND A$<>"." AND A=0
   LET A$=CHR$ (B+16)
     : IF A$="=" LET A$=","
28 IF A=1 AND B=31 GOTO 3000
29 IF B=31 THEN 50
30 IF B=11 OR B=10 GOTO 2000
33 IF A=1 LET A=0
34 C=C+1
     : CURSOR C
     : PRINT A$;CHR$ 127
35 X=X+1,H$(0)=H$(0)+A$
     : IF X>18 GOSUB 1000
40 GOTO 20
50 "Z" FOR Q=0 TO P-1
     : LPRINT W$(Q)
     : NEXT Q
     : LPRINT H$(0)
999 END
1000 X=20
1010 X=X-1
     : IF MID$ (H$(0),X,1)<>" " THEN 1010
1020 W$(P)=LEFT$ (H$(0),X-1),H$(0)=RIGHT$ (H$(0),
   LEN H$(0)-X),P=P+1,X=LEN H$(0)+1,C=X-2
1030 CLS
     : PRINT H$(0);CHR$ 127
     : CURSOR 23
     : PRINT P+1
     : RETURN
2000 IF B=11 AND P>0 LET P=P-1
2020 IF B=10 AND P<50 LET P=P+1
2030 CLS
     : H$(0)=W$(P),C=LEN W$(P),X=C
     : PRINT H$(0);CHR$ 127
     : CURSOR 23
     : PRINT P+1
     : GOTO 20
3000 "S"INPUT "Name: ";N$,"Save or Load? ";Q$
3010 IF Q$="S" PRINT #N$;W$(*)
3020 IF Q$="L" INPUT #N$;W$(*)
```

Locksmith's Master-Keying Program

H. B. Hebb
77 Ramey Ave.
Port Colborne, Ont. Canada L3K 2L2

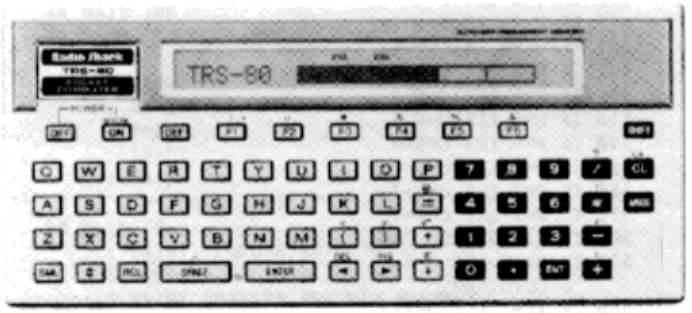
The number of requests we have received for copies of our special master-keying programs leads us to believe that there must be a remarkable number of locksmiths out there using TRS-80 computers. We use eight TRS-80 computers in our business, ranging from the PC-1 to the Model 4 with 128K. We believe the enclosed program, designed in this case for the PC-2, will be of interest to locksmiths, particularly those who receive *TRS-80 Microcomputer News*.

There are many master-keying programs available, but none, as far as we can determine, for the unique requirements of master-keying old and worn cylinders. The merits of the enclosed program will be apparent to locksmiths. It is written in the most basic-BASIC we can devise so that any user with a rudimentary knowledge of BASIC can change the program to suit his own specific requirements. This version provides locksmiths with a choice of 8 of the most common master-keying options. With minor changes the program should run on any TRS-80 equipment.

```
1 "C"BEEP 5
     : CSIZE 1
     : COLOR 0
2 WAIT 50
     : PRINT "MASTER KEY PROGRAMS"
3 WAIT 100
     : PRINT "(c) COPYRIGHT 1983"
4 WAIT 100
     : PRINT "HARRY B. HEBB"
6 WAIT 100
     : PRINT "PORT COLBORNE, ONTARIO"
8 WAIT 200
     : PRINT "OPTION 1-4 PIN, NO SUBMASTERS"
10 WAIT 200
     : PRINT "OPTION 2-4 PIN, 3 SUBMASTERS"
12 WAIT 200
     : PRINT "OPTION 3-5 PIN, NO SUBMASTERS"
14 WAIT 200
     : PRINT "OPTION 4-5 PIN, 3 SUBMASTERS"
16 WAIT 200
     : PRINT "OPTION 5-5 PIN, 6 SUBMASTERS"
18 WAIT 200
     : PRINT "OPTION 6-6 PIN, NO SUBMASTERS"
20 WAIT 200
     : PRINT "OPTION 7-6 PIN, 3 SUBMASTERS"
22 WAIT 200
     : PRINT "OPTION 8-6 PIN, 6 SUBMASTERS"
24 WAIT 100
26 INPUT "SELECT OPTION 1-8 ";O1
30 DT=-1
     : CT=-1
     : CO=0
     : K=0
     : L$=""
     : ER=0
40 IF O1=9 GOTO 1995
50 WAIT 50
     : PRINT "PLEASE ENTER"
51 WAIT 90
     : PRINT "MASTER KEY PIN NUMBERS"
53 WAIT 90
     : PRINT "MUST NOT BE GREATER THAN 9"
```

```
6Ø INPUT "PIN NO 1 ";T
7Ø INPUT "PIN NO 2 ";U
8Ø INPUT "PIN NO 3 ";V
9Ø INPUT "PIN NO 4 ";W
1ØØ IF O1<=2 GOTO 14Ø
11Ø INPUT "PIN NO 5 ";X
12Ø IF O1<=5 GOTO 14Ø
13Ø INPUT "PIN NO 6 ";Y
14Ø COLOR 3
     :  ON O1 GOTO 15Ø,16Ø,17Ø,18Ø,19Ø,2ØØ,21Ø,22Ø
15Ø WAIT 5
     : LPRINT T;U;V;W
     : GOTO 23Ø
16Ø WAIT 5
     : LPRINT T;U;V;W
     : GOTO 23Ø
17Ø WAIT 5
     : LPRINT T;U;V;W;X
     : GOTO 23Ø
18Ø WAIT 5
     : LPRINT T;U;V;W;X
     : GOTO 23Ø
19Ø WAIT 5
     : LPRINT T;U;V;W;X
     : GOTO 23Ø
2ØØ WAIT 5
     : LPRINT T;U;V;W;X;Y
     : GOTO 23Ø
21Ø WAIT 5
     : LPRINT T;U;V;W;X;Y
     : GOTO 23Ø
22Ø WAIT 5
     : LPRINT T;U;V;W;X;Y
     : GOTO 23Ø
23Ø COLOR Ø
     : IF O1=2 OR O1=4 OR O1=5 OR O1=7 OR O1=8
     LET Q=T
     : GOTO 32Ø
24Ø IF T<=2 LET Q=T+3
25Ø IF T=3 LET Q=T-3
26Ø IF T=4 LET Q=T-3
27Ø IF T=5 LET Q=T-5
28Ø IF T=6 LET Q=T-5
29Ø IF T=7 LET Q=T-7
3ØØ IF T=8 LET Q=T-7
31Ø IF T=9 LET Q=T-9
32Ø IF U<=2 LET A=U+3
33Ø IF U=3 LET A=U-3
34Ø IF U=4 LET A=U-3
35Ø IF U=5 LET A=U-5
36Ø IF U=6 LET A=U-5
37Ø IF U=7 LET A=U-7
38Ø IF U=8 LET A=U-7
39Ø IF U=9 LET A=U-9
4ØØ IF V<=2 LET B=V+3
41Ø IF V=3 LET B=V-3
42Ø IF V=4 LET B=V-3
43Ø IF V=5 LET B=V-5
44Ø IF V=6 LET B=V-5
45Ø IF V=7 LET B=V-7
46Ø IF V=8 LET B=V-7
47Ø IF V=9 LET B=V-9
48Ø IF W<=2 LET C=W+3
49Ø IF W=3 LET C=W-3
5ØØ IF W=4 LET C=W-3
51Ø IF W=5 LET C=W-5
52Ø IF W=6 LET C=W-5
53Ø IF W=7 LET C=W-7
54Ø IF W=8 LET C=W-7
55Ø IF W=9 LET C=W-9
56Ø IF O1<=2 GOTO 78Ø
57Ø IF O1=5 LET D=X
     : GOTO 66Ø
58Ø IF X<=2 LET D=X+3
59Ø IF X=3 LET D=X-3
6ØØ IF X=4 LET D=X-3
61Ø IF X=5 LET D=X-5
62Ø IF X=6 LET D=X-5
63Ø IF X=7 LET D=X-7
64Ø IF X=8 LET D=X-7
65Ø IF X=9 LET D=X-9
66Ø IF O1<=5 GOTO 77Ø
67Ø IF O1=8 LET E=Y
     : GOTO 76Ø
68Ø IF Y<=2 LET E=Y+3
69Ø IF Y=3 LET E=Y-3
7ØØ IF Y=4 LET E=Y-3
71Ø IF Y=5 LET E=Y-5
72Ø IF Y=6 LET E=Y-5
73Ø IF Y=7 LET E=Y-7
74Ø IF Y=8 LET E=Y-7
75Ø IF Y=9 LET E=Y-9
76Ø IF E>9 GOTO 11ØØ
77Ø IF D>9 GOTO 128Ø
78Ø IF C>9 GOTO 145Ø
79Ø IF B>9 GOTO 162Ø
8ØØ IF A>9 GOTO 179Ø
81Ø IF CO=Ø GOTO 84Ø
82Ø IF O1=2 OR O1=4 OR O1=5 OR O1=7 OR O1=8 GOTO 198Ø
83Ø IF Q>9 GOTO 198Ø
84Ø K=K+1
     : L$=""
85Ø ON O1 GOTO 86Ø,87Ø,88Ø,89Ø,9ØØ,92Ø,93Ø,94Ø
86Ø WAIT 1Ø
     : LPRINT Q;A;B;C;L$
     : GOTO 96Ø
87Ø WAIT 1Ø
     : LPRINT T;A;B;C;L$
     : GOTO 96Ø
88Ø WAIT 1Ø
     : LPRINT Q;A;B;C;D;L$
     : GOTO 96Ø
89Ø WAIT 1Ø
     : LPRINT T;A;B;C;D;L$
     : GOTO 96Ø
9ØØ IF CT=C GOTO 96Ø
91Ø WAIT 1Ø
     : LPRINT T;A;B;C;X;L$
     : LET CT=C
     : GOTO 96Ø
92Ø WAIT 1Ø
     : LPRINT Q;A;B;C;D;E;L$
     : GOTO 96Ø
93Ø WAIT 1Ø
     : LPRINT T;A;B;C;D;E;L$
     : GOTO 96Ø
94Ø IF DT=D GOTO 96Ø
95Ø WAIT 1Ø
     : LPRINT T;A;B;C;D;Y;L$
     : GOTO 96Ø
96Ø ON O1 GOTO 133Ø,133Ø,115Ø,115Ø,115Ø,97Ø,97Ø,97Ø
97Ø IF O1=8 GOTO 115Ø
98Ø IF E+2=Y-1 GOTO 1Ø1Ø
99Ø IF E+2=Y+1 GOTO 1Ø1Ø
1ØØØ E=E+2
     : GOTO 76Ø
1Ø1Ø IF E+4=Y-1 GOTO 1Ø4Ø
1Ø2Ø IF E+4=Y+1 GOTO 1Ø4Ø
1Ø3Ø E=E+4
     : GOTO 76Ø
1Ø4Ø IF E+6=Y-1 GOTO 1Ø7Ø
1Ø5Ø IF E+6=Y+1 GOTO 1Ø7Ø
1Ø6Ø E=E+6
     : GOTO 76Ø
1Ø7Ø IF E+8=Y-1 GOTO 11ØØ
1Ø8Ø IF E+8=Y+1 GOTO 11ØØ
1Ø9Ø E=E+8
     : GOTO 76Ø
11ØØ IF (Y=3) OR (Y=5) OR (Y=7) OR (Y=9) LET E=Ø
     : GOTO 115Ø
111Ø IF (Y=4) OR (Y=6) OR (Y=8) LET E=1
     : GOTO 115Ø
```

```
112Ø IF Y=Ø LET E=3
     : GOTO 115Ø
113Ø IF Y=1 LET E=4
     : GOTO 115Ø
114Ø IF Y=2 LET E=5
     : GOTO 115Ø
115Ø IF O1=5 GOTO 133Ø
116Ø IF D+2=X-1 GOTO 119Ø
117Ø IF D+2=X+1 GOTO 119Ø
118Ø D=D+2
     : GOTO 77Ø
119Ø IF D+4=X-1 GOTO 122Ø
12ØØ IF D+4=X+1 GOTO 122Ø
121Ø D=D+4
     : GOTO 77Ø
122Ø IF D+6=X-1 GOTO 125Ø
123Ø IF D+6=X+1 GOTO 125Ø
124Ø D=D+6
     : GOTO 77Ø
125Ø IF D+8=X-1 GOTO 128Ø
126Ø IF D+8=X+1 GOTO 128Ø
127Ø D=D+8
     : GOTO 77Ø
128Ø IF X=3 OR X=5 OR X=7 OR X=9 LET D=Ø
     : GOTO 133Ø
129Ø IF X=4 OR X=6 OR X=8 LET D=1
     : GOTO 133Ø
13ØØ IF X=Ø LET D=3
     : GOTO 133Ø
131Ø IF X=1 LET D=4
     : GOTO 133Ø
132Ø IF X=2 LET D=5
     : GOTO 133Ø
133Ø IF C+2=W-1 GOTO 136Ø
134Ø IF C+2=W+1 GOTO 136Ø
135Ø C=C+2
     : GOTO 78Ø
136Ø IF C+4=W-1 GOTO 139Ø
137Ø IF C+4=W+1 GOTO 139Ø
138Ø C=C+4
     : GOTO 78Ø
139Ø IF C+6=W-1 GOTO 142Ø
14ØØ IF C+6=W+1 GOTO 142Ø
141Ø C=C+6
     : GOTO 78Ø
142Ø IF C+8=W-1 GOTO 145Ø
143Ø IF C+8=W+1 GOTO 145Ø
144Ø C=C+8
     : GOTO 78Ø
145Ø IF W=3 OR W=5 OR W=7 OR W=9 LET C=Ø
     : GOTO 15ØØ
146Ø IF W=4 OR W=6 OR W=8 LET C=1
     : GOTO 15ØØ
147Ø IF W=Ø LET C=3
     : GOTO 15ØØ
148Ø IF W=1 LET C=4
     : GOTO 15ØØ
149Ø IF W=2 LET C=5
     : GOTO 15ØØ
15ØØ IF B+2=V-1 GOTO 153Ø
151Ø IF B+2=V+1 GOTO 153Ø
152Ø B=B+2
     : GOTO 79Ø
153Ø IF B+4=V-1 GOTO 156Ø
154Ø IF B+4=V+1 GOTO 156Ø
155Ø B=B+4
     : GOTO 79Ø
156Ø IF B+6=V-1 GOTO 159Ø
157Ø IF B+6=V+1 GOTO 159Ø
158Ø B=B+6
     : GOTO 79Ø
159Ø IF B+8=V-1 GOTO 162Ø
16ØØ IF B+8=V+1 GOTO 162Ø
161Ø B=B+8
     : GOTO 79Ø
162Ø IF V=3 OR V=5 OR V=7 OR V=9 LET B=Ø
     : GOTO 167Ø
163Ø IF V=4 OR V=6 OR V=8 LET B=1
     : GOTO 167Ø
164Ø IF V=Ø LET B=3
     : GOTO 167Ø
165Ø IF V=1 LET B=4
     : GOTO 167Ø
166Ø IF V=2 LET B=5
     : GOTO 167Ø
167Ø IF A+2=U-1 GOTO 17ØØ
168Ø IF A+2=U+1 GOTO 17ØØ
169Ø A=A+2
     : GOTO 8ØØ
17ØØ IF A+4=U-1 GOTO 173Ø
171Ø IF A+4=U+1 GOTO 173Ø
172Ø A=A+4
     : GOTO 8ØØ
173Ø IF A+6=U-1 GOTO 176Ø
174Ø IF A+6=U+1 GOTO 176Ø
175Ø A=A+6
     : GOTO 8ØØ
176Ø IF A+8=U-1 GOTO 179Ø
177Ø IF A+8=U+1 GOTO 179Ø
178Ø A=A+8
     : GOTO 8ØØ
179Ø IF U=3 OR U=5 OR U=7 OR U=9 LET A=Ø
     : GOTO 185Ø
18ØØ IF U=4 OR U=6 OR U=8 LET A=1
     : GOTO 185Ø
181Ø IF U=Ø LET A=3
     : GOTO 185Ø
182Ø IF U=1 LET A=4
     : GOTO 185Ø
183Ø IF U=2 LET A=5
     : GOTO 185Ø
184Ø IF O1=2 OR O1=4 OR O1=5 OR O1=7
     OR O1=8 GOTO 198Ø
185Ø IF Q+2=T-1 GOTO 188Ø
186Ø IF Q+2=T+1 GOTO 188Ø
187Ø Q=Q+2
     : GOTO 82Ø
188Ø IF Q+4=T-1 GOTO 191Ø
189Ø IF Q+4=T+1 GOTO 191Ø
19ØØ Q=Q+4
     : GOTO 82Ø
191Ø IF Q+6=T-1 GOTO 194Ø
192Ø IF Q+6=T+1 GOTO 194Ø
193Ø Q=Q+6
     : GOTO 82Ø
194Ø IF Q+8=T-1 GOTO 198Ø
195Ø IF Q+8=T+1 GOTO 198Ø
196Ø LET CO=1
197Ø Q=Q+8
     : GOTO 82Ø
198Ø COLOR 2
     : WAIT 2Ø
     : LPRINT "THE NUMBER OF CHANGE KEYS =";K
2ØØØ END
```



OS9 Assembly Language

Earl Bollinger

OS9's Device Path Descriptor Modules provide users with expanded capabilities. One of OS9's many features is the modular device independent input/output system. This system is designed to be easily expanded or extended as the needs or requirements of the user change over time. The purpose of this article is to discuss the Device Path Descriptor Module, specifically the ones used by the Sequential Character File Manager, in more detail than that provided by the manuals. More detailed information can be obtained from the OS9 Technical Information manual in chapter six.

OS9 uses a module concept in programming. All programs used with OS9 are normally considered to be modules. A modular concept allows you to easily add, delete or update modules within the OS9 system— within reason, of course. Every module within OS9 is relatively independent of the other modules. Thus, any changes or additions to one module will have little or no effect on the other modules. It is this fact that gives OS9 its tremendous versatility and expandability as a disk operating system.

The Sequential Character File Manager (SCF) controls all the input/output devices that communicate to OS9 on a character by character basis. Devices such as a modem, terminal, or a printer must have all their data sent via the Sequential Character File Manager module (SCFMAN). In order to accomplish this task, SCFMAN uses Path Descriptor Modules. Path Descriptor Modules provide pertinent information about a particular device and what is needed to input or output a byte or a character via that device.

The device path descriptor modules provide OS9 with a device independent I/O system. The actual details or requirements for sending or receiving data is left to the actual device driver module which is defined by the path descriptor. SCFMAN does not have to worry about the details of the data transfer; it simply manages the data. This allows you to have several path descriptor modules, each with their own set of options for a common device driver. Since device descriptors occupy less memory than device drivers, you can make more efficient use of memory by minimizing the use of device drivers.

If two or more devices can use the same device driver (e.g. a RS-232 serial port) then that is where path descriptors become useful. For example, in this case let's say we have two printers that we can use with our Color Computer. Printer A requires that we send data to it at a 1200 baud rate with carriage returns only (no line feeds). Printer B will only work at 300 baud data rate and needs both carriage returns and line feeds. Both printers use the data format provided by the PRINTER device driver (8 data bits with two stop bits) which comes with OS9. All we need now are two special device path descriptors defined for handling the two different printers.

To be usable by OS9, the path descriptor module must be properly designed and configured; otherwise OS9 will refuse to load and use it. It must have a module header with a header parity check, a module body, and a cyclic redundancy checksum (CRC) at the end of the entire module. Having to perform parity and CRC calculations would be a tremendous chore, so the OS9 Assembler is designed to handle this task for the user.

At first, it appears that a header and a CRC value would be an inconvenience in programming. But OS9 is designed to be a multiuser/multitasking system. Any module or program that may be bad or damaged would cause the entire system to fail or crash. Thus, OS9 will refuse to execute any program that does not have a proper header and CRC value. The header also gives OS9 important information about a module prior to execution. This information would include the type of module, how much memory to allocate to the module for execution, any additional support modules that may be required, and so on.

Program example one is what an assembly language source program for a path descriptor looks like when it is entered into memory using the OS9 Macro Text Editor. Although the program looks somewhat unattractive in this form, when it is assembled by the OS9 Assembler, it becomes a neatly formatted assembly listing. Because you are limited to a 16 by 32 character screen display for entering and editing source programs, you cannot really take advantage of prettier input methods.

Program example two demonstrates what the program looks like after the OS9 Assembler has assembled it.

A path descriptor module must abide by all the OS9 rules pertaining to structure. Therefore, it has a header, a body and a CRC value at the end. The actual header is set up by the assembler when the pseudo-opcode "MOD" is used at the beginning of the program. After "MOD" comes the parameters that the assembler requires to set up a proper header for OS9. The DVEND label tells the assembler where the actual end of the program is for the CRC calculations. DVNAM identifies the proper path descriptor name to be used in the header. TYPE is used to identify the type or kind of program for OS9. REENT + 1 is used to let OS9 know that this program is usable in a multiuser environment. SCFNAM identifies which I/O Manager module is to be used with the path descriptor. DRVNAM identifies which device driver will be used with the path descriptor for the actual data transfers that will take place. All of these items are defined within the program as labels or symbols. The actual structure for a path descriptor module must remain as shown in the two program examples, between the MOD and EMOD pseudo-opcodes.

Program #1

```
**********************************
*                                *
* SERIAL PORT DEVICE DESCRIPTOR *
*                                *
* THIS PROGRAM DEMONSTRATES THE *
* NORMAL ASSEMBLY LANGUAGE       *
* STRUCTURE FOR A SERIAL PORT    *
* PATH DESCRIPTOR MODULE AS USED*
* WITH THE OS-9 DISK OPERATING   *
* SYSTEM.                        *
**********************************
*
UPDATE SET Ø3
DEVICE SET $FØ
OBJECT SET $Ø1
REENT SET $8Ø
*
* MODULE HEADER INFORMATION
*
TYPE SET DEVICE+OBJECT
 MOD DVEND,DVNAM,TYPE,REENT+1,SCFNAM,DRVNAM
 FCB UPDATE
 FCB $FF,$ØØ,$ØØ ADDRESS OF SERIAL PORT
 FCB OPTSIZ
OPTBGN EQU *
*
* PATH DESCRIPTOR OPTION LIST
*
CLASS FCB ØØ SCF TYPE DESIGNATION
UPC FCB ØØ UPPER/LOWER CASE
BSE FCB Ø1 BACKSPACE ECHO BS,SP,BS
DCLRL FCB ØØ DELETE OR CLR LINE
ECHO FCB Ø1 ECHO CHARS
AUTOLF FCB ØØ AUTO LINE FEED
NULL FCB ØØ NO END OF LINE NULLS
PAUSE FCB ØØ END OF LINE PAUSE OFF
PAGE FCB 16 LINES PER PAGE
BKSPC FCB Ø8 BACKSPACE CHAR
DELETE FCB 24 DELETE LINE CHAR
EOFREC FCB 13 END OF RECORD (LINE)
ENDFIL FCB 27 END OF FILE CHAR
RPRLL FCB Ø4 REPRINT LAST LINE
DUPE FCB Ø1 DUPLICATE LAST LINE
PSCHAR FCB 23 PAUSE CHAR
INTRUP FCB Ø3 KEYBOARD INTERRUPT CHAR
ABORT FCB $15 KEYBOARD ABORT CHAR
BKSPE FCB Ø8 BACKSOACE ECHO CHAR
OVRFLW FCB Ø7 BELL-LINE OVERFLOW CHAR
DVINIT FCB ØØ DEVICE INITIALIZATION VALUE
BAUD FCB Ø2 BAUD RATE VALUE
DV2 FDB DVNAM OFFSET TO ATTACHED DEVICE NAME
STATUS FDB ØØ OFFSET TO DEVICE STATUS ROUTINE
ERROR FCB ØØ I/O ERROR CODE STATUS
OPTSIZ EQU *-OPTBGN
*
*
DVNAM FCS /P2/ DEVICE PATH NAME
```

```
                    SCFNAM FCS /SCF/ DEVICE MANAGER NAME
                    DRVNAM FCS /PRINTER/ SERIAL PORT DRIVER NAME
                     EMOD MODULE CRC VALUE
                    DVEND EQU *
                     END
```

Program #2

```
ØØØØ1          **********************************
ØØØØ2          *                                *
ØØØØ3          * SERIAL PORT DEVICE DESCRIPTOR *
ØØØØ4          *                                *
ØØØØ5          * THIS PROGRAM DEMONSTRATES THE *
ØØØØ6          * NORMAL ASSEMBLY LANGUAGE      *
ØØØØ7          * STRUCTURE FOR A SERIAL PORT   *
ØØØØ8          * PATH DESCRIPTOR MODULE AS USED*
ØØØØ9          * WITH THE OS-9 DISK OPERATING  *
ØØØ1Ø          * SYSTEM.                       *
ØØØ11          **********************************
ØØØ12          *
ØØØ13    ØØØ3            UPDATE   SET   Ø3
ØØØ14    ØØFØ            DEVICE   SET   $FØ
ØØØ15    ØØØ1            OBJECT   SET   $Ø1
ØØØ16    ØØ8Ø            REENT    SET   $8Ø
ØØØ17          *
ØØØ18          * MODULE HEADER INFORMATION
ØØØ19          *
ØØØ2Ø    ØØF1            TYPE     SET   DEVICE+OBJECT
ØØØ21    ØØØØ 87CDØØ3C           MOD   DVEND,DVNAM,TYPE,REENT+1,SCFNAM,DRVNAM
ØØØ22    ØØØD Ø3                 FCB   UPDATE
ØØØ23    ØØØE FFØØØØ             FCB   $FF,$ØØ,$ØØ ADDRESS OF SERIAL PORT
ØØØ24    ØØ11 1B                 FCB   OPTSIZ
ØØØ25    ØØ12            OPTBGN   EQU   *
ØØØ26          *
ØØØ27          * PATH DESCRIPTOR OPTION LIST
ØØØ28          *
ØØØ29    ØØ12 ØØ         CLASS    FCB   ØØ        SCF TYPE DESIGNATION
ØØØ3Ø    ØØ13 ØØ         UPC      FCB   ØØ        UPPER/LOWER CASE
ØØØ31    ØØ14 Ø1         BSE      FCB   Ø1        BACKSPACE ECHO BS,SP,BS
ØØØ32    ØØ15 ØØ         DCLRL    FCB   ØØ        DELETE OR CLR LINE
ØØØ33    ØØ16 Ø1         ECHO     FCB   Ø1        ECHO CHARS
ØØØ34    ØØ17 ØØ         AUTOLF   FCB   ØØ        AUTO LINE FEED
ØØØ35    ØØ18 ØØ         NULL     FCB   ØØ        NO END OF LINE NULLS
ØØØ36    ØØ19 ØØ         PAUSE    FCB   ØØ        END OF LINE PAUSE OFF
ØØØ37    ØØ1A 1Ø         PAGE     FCB   16        LINES PER PAGE
ØØØ38    ØØ1B Ø8         BKSPC    FCB   Ø8        BACKSPACE CHAR
ØØØ39    ØØ1C 18         DELETE   FCB   24        DELETE LINE CHAR
ØØØ4Ø    ØØ1D ØD         EOFREC   FCB   13        END OF RECORD (LINE)
ØØØ41    ØØ1E 1B         ENDFIL   FCB   27        END OF FILE CHAR
ØØØ42    ØØ1F Ø4         RPRLL    FCB   Ø4        REPRINT LAST LINE
ØØØ43    ØØ2Ø Ø1         DUPE     FCB   Ø1        DUPLICATE LAST LINE
ØØØ44    ØØ21 17         PSCHAR   FCB   23        PAUSE CHAR
ØØØ45    ØØ22 Ø3         INTRUP   FCB   Ø3        KEYBOARD INTERRUPT CHAR
ØØØ46    ØØ23 15         ABORT    FCB   $15       KEYBOARD ABORT CHAR
ØØØ47    ØØ24 Ø8         BKSPE    FCB   Ø8        BACKSOACE ECHO CHAR
ØØØ48    ØØ25 Ø7         OVRFLW   FCB   Ø7        BELL-LINE OVERFLOW CHAR
ØØØ49    ØØ26 ØØ         DVINIT   FCB   ØØ        DEVICE INITIALIZATION VALUE
ØØØ5Ø    ØØ27 Ø2         BAUD     FCB   Ø2        BAUD RATE VALUE
ØØØ51    ØØ28 ØØ2D       DV2      FDB   DVNAM     OFFSET TO ATTACHED DEVICE NAME
```

```
00052   002A 0000       STATUS   FDB   00          OFFSET TO DEVICE STATUS ROUTIN
00053   002C 00         ERROR    FCB   00          I/O ERROR CODE STATUS
00054   001B             OPTSIZ   EQU   *-OPTBGN
00055         *
00056         *
00057   002D 50B2        DVNAM    FCS   /P2/        DEVICE PATH NAME
00058   002F 5343C6      SCFNAM   FCS   /SCF/       DEVICE MANAGER NAME
00059   0032 5052494E    DRVNAM   FCS   /PRINTER/   SERIAL PORT DRIVER NAME
00060   0039 DF07C3              EMOD               MODULE CRC VALUE
00061   003C             DVEND    EQU   *
00062                            END
00000 error(s)
00000 warning(s)
$003C 0060 program bytes generated
$0000 0000 data bytes allocated
$0250 00592 bytes used for symbols
```

Address location 0E hexadecimal contains the absolute in memory location of the serial port itself. Three bytes of memory are reserved for the absolute address of the serial port. Starting at address 12 hexadecimal and going on to 2C hexadecimal are the serial port options. These options control all the pertinent information about a particular serial port. Specific information about these options are obtained from two areas of the OS9 manuals. The first source of information is the OS9 Technical Information Manual. The options are discussed in some detail in chapter six. The second source of information is in the OS9 Commands manual under the TMODE utility command.

The options between offset 12 and 27 hexadecimal can be changed using the TMODE command. As a matter of fact, the TMODE command, as explained in OS9 Commands, is where much of the information for modifying or changing the options are obtained. Some options, as related to the Color Computer, will have little or no effect on the system if used. The DVINIT value is primarily used as the initialization or programming value for MC6850 types of ACIA devices. The Color Computer uses a MC6821 PIA device for serial communications, thereby rendering the DVINIT value meaningless. The BAUD rate value normally is only of usefulness at either a 600 or a 1200 baud rate value (2 = 600, 3 = 1200). ECHO is used primarily in a terminal mode of operation; it represents a Full Duplex or Half Duplex operation mode. EOFREC is normally a 13 hexadecimal value representing a carriage return (or ENTER) ASCII character. Variable length records normally end with a carriage return or line feed character. Variable length records are typically ASCII text files arranged as lines ending with a carriage return. INTRUP and ABORT are the two keyboard values used to interrupt or abort a particular task in process.

DVNAM contains the actual device descriptor device name (typical names are P, P1, P2, TERM, T1, T2, etc.). SCFNAM contains the name of the actual device data manager module; in this case it is SCFMAN. DRVNAM contains the name of the actual device driver module that performs the task of inputting and outputting data from the system. OS9 for the Color Computer actually has two device driver modules for the serial port. Besides the PRINTER module, there is also the RS232 module. PRINTER is used primarily as a printer driver. RS232 is used as the additional terminal driver for multiuser bidirectional applications.

This path descriptor module can be used for both the PRINTER and RS232 modules by simply changing the DRVNAM string name. You may also want to change DVNAM to a unique name. Most of the other options become useful in a terminal mode when used with the RS232 driver module. For terminals the PAUSE, ECHO, AUTOLF, BSE and so on all become more meaningful in actual operation.

Instead of using the USE /pathlist/OS9Defs command at the beginning of the program, the author simply assigned the four labels UPDATE, DEVICE, OBJECT and REENT their actual values as defined in the OS9Defs file. Use the LIST /D0/DEFS/OS9Defs >/P command to obtain a hard copy listing of this file. Normally you would use the USE pseudo-opcode command to tell the assembler to scan the OS9Defs file for predefined labels and symbols. But doing so forces you to allocate a larger (about 10k) amount of memory to the assembler for symbol table storage. In this application, the author simply defined the labels within the program in order to save on wasted memory area; especially since out of hundreds of symbols you only need four.

Finally, after entering the program, use the following command line (assuming a two drive OS9 system) to assemble it:

```
OS9:ASM /D1/Device.src L O=/D0/CMDS/PX >/P >>/P
```
ENTER

In this example, the program is named Device.src. PX is the name given to the binary object file that is created and put into the OS9 Commands execution directory. L generates a formatted output listing using >/P and >>/P to output the listing to the printer. To use or test the module use the LOAD PX command to install the module in memory. Then call it using a >/PX redirection command. If you desire, you can permanently install the module into the system using the LOAD, LINK and COBBLER commands in order to build a new system OS9Boot file.

As for the two device path descriptors discussed earlier, set up the first one's name as P1. Then set the AUTOLF label value as 01 and the BAUD rate value as 03. The second printer path descriptor can be named P2. Set up it's AUTO value as 00 and the BAUD rate as 02. After assembly you have two different path descriptors named P1 and P2. By using a LOAD P1 or P2 command, you can then use them to output data onto a printer via the serial port.

Genealogy

n A. Madera
Santa Dorotea Circle
ohnert Park, CA 94928

I would like to share my system of programs with the readers of *TRS-80 Microcomputer News*. Especially with those readers who want a very sive, but yet effective system for recording and storing their data. In return I would hope that someone can suggest how I might improve the programs to include the use of two cassette tape recorders when making corrections to the files.

The Basic program is self explanatory; it is a page by page recording of data for each individual in the file. Each segment is limited to 20 pages (1 page per person, 10 lines per page, 50 spaces per line). this is due to the size of by machine (16K BASIC). After the initial run of the program, subsequent segments are started with the command "Run 120". Running this data through the next program produces a Numerical Index which can also be used as a Pedigree Chart Expanded. The 3rd program is for correcting, editing, deletions, taping, printing, alphabetizing, etc. It is the work horse of the programs. The next program is for recording narrative data such as family histories or copies of letters, etc. (80 spaces per line). And the last program is a very helpful chart for the amateur genealogist like myself in determining the relationship of the various family members recorded in my system of filing.

For a long time I sought to find a very simple and inexpensive system for recording and storing the data I had reflected in my genealogy efforts pertaining to my 6 families. The storage in 3 ring notebooks was too voluminous, and there was too much typing involved when making copies to share with others working on the project. After an unfortunate experience with another brand, I settled on the TRS-80 Color Computer, BASIC 16K; the CGP-115 Printer; and the cassette tape system for storage. (I know, the disk system is supposed to be better, etc., but I wanted a simple and inexpensive system.) I designed these programs and they work marvelously.

As stated earlier, I hope someone can suggest an improvement to the system, especially in the CHANGE program whereby you can feed the data into the computer with one recorder, allow the changes to be made, and then feed out the corrected data into the 2nd recorder without having to constantly change tapes in the one recorder. Or, I would like to hear the critiques of the system for possible improvements. All comments are solicited.

```
1 ' BASIC PROGRAM FOR GENEALOGY FILES
2 ' BY -- ERVIN A. MADERA;
     843 SANTA DOROTEA CIRCLE;
     ROHNERT PARK; CA  94928
3 CLEAR 10000
     : CLS(3)
4 PRINT@106,"------ FAMILY";
5 FOR X=150-225
     : SOUND X,1
     : NEXT X
6 ML$="T255;V31;01;1;10;1;3;4;5;6;3;5;6;4;9;8"
     7MX$=";V25;7;8;1;2;2;6;9;6;7;5;1;2;1;;
     6;6;9;3;4;8;9;6;5;4;8;5;1;9;3;5;5;
     V20;4;6;7;5;8;9;1;4;5;4;3;4;7;5;3;4;
     9;1;2;3;4;9;2;3;
     V15;2;3;4;5;9;4;1;1;3;2;9;1;;1;5;9;
     2;1;2;9;1;2;
     V10;1;1;2;1;9;4"
8 PRINT@426, "POSITION TAPE";
9 PRINT@456, "PRESS PLAY & RECORD";
10 PRINT@484, "PRESS <ENTER> WHEN READY";
11 INPUT ""; R$
12 OPEN "O", #-1, "GENEALOGY"
15 CLS(3)
     : PRINT"INPUT YOUR FAMILY DATA  ------
       TYPE < ZZ > WHEN FINISHED"
20 INPUT "FAM #:";A$
21 INPUT "M NAME:";B$
22 INPUT "M DPOB:";C$
23 INPUT "M DPOD:";D$
24 INPUT "S NAME:";E$
25 INPUT "S DPOB:";F$
26 INPUT "S DPOD:";G$
27 INPUT "MARRIED:";H$
28 INPUT "CHLDRN:";I$
29 INPUT "--------------------";J$
30 IF A$="ZZ" THEN 50
35 PRINT#-1, A$,B$,C$,D$,E$,F$,G$,H$,I$,J$
40 GOTO 15
50 CLOSE#-1
53 PRINT#-2, "    THE     ------------     FAMILY"
54 PRINT#-2, ""
55 PRINT#-2, "DATA CONTAINED IN THE GENEALOGY FILES
     OF----------                                (AS
     OF < DATE >  )."
60 PRINT#-2, CHR$(29)
65 PRINT#-2, CHR$(17)
     : PRINT#-2, "**********************************"
70 PRINT#-2, "1. FAMILY NUMBER"
75 PRINT#-2, "2. NAME OF FAMILY MEMBER"
80 PRINT#-2, "3. DATE/PLACE OF BIRTH OF MEMBER"
85 PRINT#-2, "4. DATE/PLACE OF DEATH OF MEMBER"
90 PRINT#-2, "5. NAME OF SPOUSE"
95 PRINT#-2, "6. DATE/PLACE OF BIRTH OF SPOUSE"
100 PRINT#-2, "7. DATE/PLACE OF DEATH OF SPOUSE"
105 PRINT#-2, "8. DATE/PLACE MARRIED"
110 PRINT#-2, "9. CHILDREN"
115 PRINT#-2, "**********************************"
116 PRINT#-2, CHR$(29)
120 CLS
     : CLS(3)
     : PRINT "REWIND THE RECORDER & PRESS PLAY";
122 INPUT "PRESS  <  ENTER  >  WHEN READY"; R$
124 OPEN "I", #-1,  "GENEALOGY"
125 IF EOF(-1) THEN 240
130 INPUT #-1, A$, B$,C$,D$,E$,F$,G$,H$,I$,J$
135 PRINT "FAM  #:" A$
140 PRINT "M NAME:" B$
145 PRINT "M DPOB:" C$
150 PRINT "M DPOD:" D$
155 PRINT "S NAME:" E$
160 PRINT "S DPOB:" F$
165 PRINT "S DPOD:" G$
170 PRINT "MARRIED:"H$
175 PRINT "CHLDRN:" I$
176 PRINT "--------------------" J$
180 PRINT#-2, "1. " A$
185 PRINT#-2, "2. " B$
190 PRINT#-2, "3. " C$
195 PRINT#-2, "4. " D$
200 PRINT#-2, "5. " E$
205 PRINT#-2, "6. " F$
210 PRINT#-2, "7. " G$
215 PRINT#-2, "8. " H$
220 PRINT#-2, "9. " I$
222 PRINT#-2, "10. -----------------" J$
225 PRINT#-2, CHR$(29)
230 GOTO 125
240 CLOSE #-1
```

```
1 'NUMERICAL INDEX PROGRAM
2 ' BY  ERVIN A. MADERA;
     843 SANTA DOROTEA CIRCLE;
     ROHNERT PARK; CA  94928
4 CLEAR 10000
     : DIM S$(201)
     : CLS(3)
6 PRINT@101, "NUMERICAL INDEX PROGRAM";
7 PRINT@426, "POSITION TAPE";
8 PRINT@456, "PRESS PLAY & RECORD";
9 PRINT@484, "PRESS <ENTER> WHEN READY";
     : INPUT R$
10 OPEN "O",#-1, "GENEALOGY"
15 CLS
     : CLS(3)
     : PRINT"INPUT YOUR FAMILY DATA  --"
16 PRINT "TYPE < ZZ > WHEN FINISHED"
20 INPUT "FAM #:";A$
21 INPUT "M NAME:";B$
22 INPUT "M DPOB:";C$
23 INPUT "M DPOD:";D$
24 INPUT "S NAME:";E$
25 INPUT "S DPOB:";F$
26 INPUT "S DPOD:";G$
27 INPUT "MARRIED:";H$
28 INPUT "CHLDRN:";I$
29 INPUT "--------------------";J$
30 IF A$="ZZ" THEN 50
35 PRINT#-1, A$,B$,C$,D$,E$,F$,G$,H$,I$,J$
40 GOTO 15
50 CLOSE #-1
55 PRINT#-2, "NUMERICAL INDEX FOR THE DATA CONTAINED
     IN THE GENEALOGY FILES OF ----------------------
     ( AS OF  <DATE>  ).
60 PRINT#-2, CHR$(29)
65 PRINT#-2, CHR$(17)
     : PRINT#-2, "**********************************"
70 PRINT#-2, ""
75 PRINT#-2, "NUMBER"TAB(16)"MEMBER" TAB(56)"SPOUSE"
80 PRINT#-2, ""
115 PRINT#-2, "**************************************"
116 PRINT#-2, CHR$(29)
120 CLS
     : CLS(3)
     : PRINT "REWIND THE RECORDER & PRESS PLAY"
122 INPUT "PRESS < ENTER > WHEN READY"; R$
124 OPEN "I", #-1,  "GENEALOGY"
125 IF EOF(-1) THEN 240
130 INPUT #-1, A$, B$,C$,D$,E$,F$,G$,H$,I$,J$
135 PRINT,TAB(5) A$ TAB(32) B$ TAB(64) E$
180 PRINT#-2, A$ TAB(16) B$ TAB(56) E$
230 GOTO 125
240 CLOSE #-1
```

```
1 '  PROGRAM FOR CHANGING OF GENEALOGY FILES
2 CLEAR 10000
     : DIM S$(201)
     : CLS(3)
3 PRINT@106, "------  FAMILY";
8 PRINT@484, "PRESS <ENTER> WHEN READY";
     : INPUT R$
10 '       GENEALOGY
20 CLS
     : CLS(3)
30 PRINT@71, "DO YOU WANT TO -----";
40 PRINT@134, "(1) INPUT ITEMS";
50 PRINT@166, "(2) REPLACE ITEMS";
60 PRINT@198, "(3) ADD TO THE LIST";
70 PRINT@230, "(4) DELETE ITEMS";
80 PRINT@262, "(5) DISPLAY ITEMS";
90 PRINT@294, "(6) SAVE ITEMS ON TAPE";
100 PRINT@326, "(7) LOAD ITEMS FROM TAPE";
105 PRINT@358, "(8) PRINT ON PRINTER";
107 PRINT@390, "(9) ALPHABETIZE DATA";
110 PRINT@458, "(1-9)";
120 INPUT M
130 IF M < 0 OR M > 9 THEN 10
140 ON M GOSUB 1000,2000,1020,3000,4000,5000,6000,
     7000,8000
150 GOTO 10
900 '
1000 '       INPUT/ADD ITEMS
1010 Y=1
1020 CLS
     : PRINT@40, "INPUT/ADD ITEMS"
1030 PRINT@101, "TYPE <ZZ> WHEN FINISHED"
1035 PRINT@165, "POSITION TAPE ON RECORDER"
1036 PRINT@231, "PRESS PLAY & RECORD"
1037 OPEN "O", #-1, "GENEALOGY"
1040 PRINT
     : PRINT "ITEM" Y;
1045 INPUT S$(Y)
1050 IF S$(Y)="ZZ" THEN 1090
1060 Y=Y+1
1070 GOTO 1040
1090 CLOSE #-1
1095 RETURN
1900 '
2000 '      REPLACE ITEMS
2005 N=0
2010 CLS
     : PRINT@34, "PRESS <ENTER> WHEN FINISHED"
2020 PRINT
     : INPUT "ITEM NO. TO REPLACE"; N
2040 IF N=0 THEN RETURN
2050 INPUT "REPLACEMENT ITEM"; S$(N)
2060 GOTO 2000
2900 '
3000 '      DELETE ITEMS
3005 N=0
3010 CLS
     : PRINT@9, "DELETE ITEMS"
3020 PRINT@34, "PRESS <ENTER> WHEN FINISHED"
3030 PRINT
     : INPUT "ITEM TO DELETE"; N
3035 IF N > Y-1 THEN 3030
3040 IF N=0 THEN RETURN
3050 FOR X=N TO Y-2
3060 S$(X) = S$(X+1)
3070 NEXT X
3080 S$(X) = ""
3090 Y = Y-1
3100 GOTO 3000
3900 '
4000 '      DISPLAY ITEMS
4010 FOR X=1 TO Y-1 STEP 10
4020 FOR Z=X TO X+9
4030 PRINT Z; S$(Z)
4040 NEXT Z
4050 INPUT "PRESS < ENTER > TO CONTINUE"; L$
4060 NEXT X
4070 RETURN
4900 '
5000 '      SAVE ITEMS ON TAPE
5010 CLS
     : PRINT@135, "SAVE ITEMS ON TAPE"
5020 PRINT@234, "POSITION TAPE"
5030 PRINT@294, "PRESS PLAY AND RECORD"
5040 PRINT@388, "PRESS <ENTER> WHEN READY"
5050 INPUT R$
5060 OPEN "O", #-1, "GENEALOGY"
5070 FOR X = 1 TO Y-1
5080 PRINT#-1, S$(X)
5090 NEXT X
5100 CLOSE#-1
     : RETURN
5900 '
6000 '      LOAD ITEMS FROM TAPE
6010 CLS
     : PRINT@136, "LOAD ITEMS FROM TAPE"
```

```
6020 PRINT@235, "REWIND TAPE"
6030 PRINT@300, "PRESS PLAY"
6040 PRINT@388, "PRESS <ENTER> WHEN READY"
6050 INPUT R$
6060 OPEN "I", #-1, "GENEALOGY"
6070 Y = 1
6080 IF EOF(-1) THEN 6120
6090 INPUT #-1, S$(Y)
6095 PRINT S$(Y)
6100 Y = Y+1
6110 GOTO 6080
6120 CLOSE #-1
     : RETURN
6900 '
7000 '       PRINT ITEMS ON PRINTER
7010 FOR X=1 TO Y-1 STEP 201
7020 FOR Z=X TO X+200
7025 PRINT, Z; S$(Z)
7030 PRINT#-2, Z; S$(Z)
7040 NEXT Z
7050 INPUT "PRESS < ENTER > TO CONTINUE"; L$
7060 NEXT X
7070 RETURN
7900 '
8000 '        ALPHABETIZE DATA
8001 B=1
8002 B=B+1
8003 PRINT
8005 FOR X=1 TO 201
8010 PRINT S$(X)
8020 NEXT X
8025 PRINT
8030 CLS
     : PRINT "PRESS PLAY & RECORD"
     : INPUT "PRESS < ENTER > WHEN READY"; R$
8035 OPEN "O", #-1, "GENEALOGY"
8055 '
8060 '        SORT ROUTINE
8070 S=1
8080 X=0
8090 X=X+1
8100 IF X>201 THEN CLOSE #-1
     : GOTO 30
8105 IF S$(X) = "ZZ" THEN 8090
8110 FOR Y=1 TO 201
8120 IF S$(Y) < S$(X) THEN X=Y
8130 NEXT Y
8140 PRINT S$(X)
8150 PRINT#-2, S$(X)
8160 PRINT#-1, S$(X)
8165 S$(X) = "ZZ"
8170 GOTO 8080
```

```
1 ' PROGRAM FOR FAMILY HISTORY NARRATIVES
2 ' BY    ERVIN A. MADERA;
     843 SANTA DOROTEA CIRCLE;
     ROHNERT PARK; CA   94928
4 CLEAR 10000
     : CLS(3)
5 PRINT@106, "--------  FAMILY";
6 ML$="T255;V31;01;1;10;1;3;4;5;6;3;5;6;4;9;8"
7 MX$=";V25;7;8;1;2;2;6;9;6;7;5;1;2;1;;
     6;6;9;3;4;8;9;6;5;4;8;5;1;9;3;5;5;
     V20;4;6;7;5;8;9;1;4;5;4;3;4;7;5;3;
     4;9;1;2;3;4;9;2;3;
     V15;2;3;4;5;9;4;1;1;3;2;9;1;;1;5;9;
     2;1;2;9;1;2;
     V10;1;1;2;1;9;4"
8 PRINT@426, "POSITION TAPE";
9 PRINT@456, "PRESS PLAY & RECORD";
10 PRINT@484, "PRESS <ENTER> WHEN READY";
11 INPUT ""; R$
12 OPEN "O", #-1, "GENEALOGY"
15 CLS(3)
     : PRINT "INPUT YOUR DATA -- TYPE <ZZ>
     WHEN FINISHED"
20 INPUT A$
30 IF A$="ZZ" THEN 50
35 PRINT#-1, A$
40 GOTO 15
50 CLOSE #-1
120 CLS
     : CLS(3)
     : PRINT "REWIND THE RECORDER & PRESS PLAY"
122 INPUT "PRESS < ENTER > WHEN READY"; R$
124 OPEN "I", #-1, "GENEALOGY"
125 IF EOF(-1) THEN 240
130 INPUT #-1,A$
135 PRINT A$
140 PRINT#-2, A$
230 GOTO 125
240 CLOSE#-1
```

```
1 '   PROGRAM FOR PRINTING RELATIONSHIP CHART
2 '  BY   ERVIN A. MADERA;
     843 SANTA DOROTEA CIRCLE;
     ROHNERT PARK; CA  94928
5 CLEAR 10000
     : CLS(3)
7 PRINT#-2, "CHART FOR DETERMINING RELATIONSHIPS
     OF FAMILY MEMBERS"
8 PRINT#-2, ""
     : PRINT#-2, ""
9 PRINT#-2, ""
     : PRINT#-2, ""
10 PRINT#-2, "X"TAB(6)"1"TAB(19)"2"TAB(32)"3"
     TAB(45)"4"TAB(57)"5"TAB(70)"6"
14 PRINT#-2, ""
15 PRINT#-2, ""
20 PRINT#-2, "A"TAB(6)"BROTHER/"TAB(19)"AUNT/UNCLE"
     TAB(32)"GREAT"TAB(45)"GREAT,G"TAB(57)"GGG"
     TAB(70)"GGGG"
25 PRINT#-2,TAB(6)"SISTER"TAB(19)"NIECE/NEPH"
     TAB(32)"AUNT/UNCLE"TAB(45)"AUNT/UNCLE"
     TAB(57)"AUNT/UNCLE"TAB(70)"AUNT/UNCLE"
28 PRINT#-2,TAB(32)"NIECE/NEPH"TAB(44)"NIECE/NEPH"
     TAB(56)"NIECE/NEPH"TAB(68)"NIECE/NEPH"
30 PRINT#-2, ""
35 PRINT#-2, ""
40 PRINT#-2, "B"TAB(6)"AUNT/UNCLE"TAB(32)"1ST COUSIN"
     TAB(45)"1ST COUSIN"TAB(57)"1ST COUSIN"
     TAB(70)"1ST COUSIN"
45 PRINT#-2,TAB(6)"NIECE/NEPH"TAB(19)"1ST COUSIN"
     TAB(32)"ONCE"TAB(45)"TWICE"TAB(57)"3 TIMES"
     TAB(70)"4 TIMES"
50 PRINT#-2,TAB(32)"REMOVED"TAB(45)"REMOVED"
     TAB(57)"REMOVED"TAB(70)"REMOVED"
55 PRINT#-2, ""
60 PRINT#-2, ""
65 PRINT#-2, "C"TAB(6)"GREAT"TAB(19)"1ST COUSIN"
     TAB(45)"2ND COUSIN"TAB(57)"2ND COUSIN"
     TAB(70)"2ND COUSIN"
70 PRINT#-2,TAB(6)"AUNT/UNCLE"TAB(19)"ONCE"
     TAB(32)"2ND COUSIN"TAB(45)"ONCE"
     TAB(57)"TWICE"TAB(70)"3 TIMES"
75 PRINT#-2,TAB(6)"NIECE/NEPH"TAB(19)"REMOVED"
     TAB(45)"REMOVED"TAB(57)"REMOVED"TAB(70)"REMOVED"
80 PRINT#-2, ""
85 PRINT#-2, ""
90 PRINT#-2, "D"TAB(6)"GREAT,G"TAB(19)"1ST COUSIN"
     TAB(32)"2ND COUSIN"TAB(57)"3RD COUSIN"
     TAB(70)"3RD COUSIN"
95 PRINT#-2,TAB(6)"AUNT/UNCLE"TAB(19)"TWICE"
     TAB(32)"ONCE"TAB(45)"3RD COUSIN"
     TAB(57)"ONCE"TAB(70)"TWICE"
100 PRINT#-2,TAB(6)"NIECE/NEPH"TAB(19)"REMOVED"
     TAB(32)"REMOVED"TAB(57)"REMOVED"TAB(70)"REMOVED"
105 PRINT#-2, ""
110 PRINT#-2, ""
```

```
115 PRINT#-2, "E"TAB(6)"GGG"TAB(19)"1ST COUSIN"
    TAB(32)"2ND COUSIN"TAB(45)"3RD COUSIN"
    TAB(70)"4TH COUSIN"
120 PRINT#-2,TAB(6)"AUNT/UNCLE"TAB(19)"3 TIMES"
    TAB(32)"TWICE"TAB(45)"ONCE"TAB(57)"4TH COUSIN"
    TAB(70)"ONCE"
125 PRINT#-2,TAB(6)"NIECE/NEPH"TAB(19)"REMOVED"
    TAB(32)"REMOVED"TAB(45)"REMOVED"TAB(70)"REMOVED"
130 PRINT#-2, ""
135 PRINT#-2, ""
140 PRINT#-2, "F"TAB(6)"GGGG"TAB(19)"1ST COUSIN"
    TAB(32)"2ND COUSIN"TAB(45)"3RD COUSIN"
    TAB(57)"4TH COUSIN"
145 PRINT#-2,TAB(6)"AUNT/UNCLE"TAB(19)"4 TIMES"
    TAB(32)"3 TIMES"TAB(45)"TWICE"TAB(57)"ONCE"
    TAB(70)"5TH COUSIN"
150 PRINT#-2,TAB(6)"NIECE/NEPH"TAB(19)"REMOVED"
    TAB(32)"REMOVED"TAB(45)"REMOVED"TAB(57)"REMOVED"
155 PRINT#-2, ""
    : PRINT#-2, ""
160 PRINT#-2, ""
    : PRINT#-2, ""
170 PRINT#-2, "TO USE THE CHART, DETERMINE THE COMMON
    ANCESTOR.  THEN COUNT ACROSS THE TOP"
172 PRINT#-2, "(X-6) TO DETERMINE IN WHICH GENERATION
    FROM THAT ANCESTOR YOU ARE.  FOR THE"
174 PRINT#-2, "OTHER PERSON, COUNT THE SAME WAY DOWN
    (X-F) TO DETERMINE WHAT GENERATION "
180 PRINT#-2, "FROM THE COMMON ANCESTOR HE WOULD BE.
    MOVING RIGHT ALONG THE LINE SELECTED"
182 PRINT#-2, "FROM X TO F; AND MOVING DOWN ALONG THE
    COLUMN SELECTED FROM X TO 6; YOU WILL"
184 PRINT#-2, "FIND THE SQUARE INDICATING THE
    RELATIONSHIP OF THE 2 PERSONS.  FOR EXAMPLE;"
185 PRINT#-2, "IF ONE PERSON IS IN THE 3RD GENERATION
    AND THE OTHER PERSON IS IN THE 5TH"
187 PRINT#-2, "GENERATION OF A COMMON ANCESTOR; THEY
    ARE 2ND COUSINS TWICE REMOVED."
```

Calculating Interest on the Color Computer

Clifford D. Jensen
1925 West 10th Ave.
Apache Junction, AZ 85220

This interest calculating program was programmed for 16K extended Color BASIC. It computes for both compound and simple interest, displays the results on the screen, and then provides the option of obtaining a hard copy if desired.

The program is completely menu driven and prompts the user for all required inputs. It is this ease of use and of output readability that makes the program so lengthy.

If simple interest is selected, the program provides the option of selecting either days or years for the term. If the days option is selected, it further provides for either entering the number of days directly or entering the beginning date and ending date and letting the computer calculate the number of days. If beginning and ending dates are entered, it then displays both dates with the month, day, year, day of the week, and numeric number of day of year, and the number of days between. The program then uses this number of days to perform the computations without further input from the user.

```
10 DIM A$(50)
20 CLS
    : PRINT"       INTEREST CALCULATOR"
30 FOR X=1 TO 460*2
    : NEXT X
40 CLS
    : PRINT "DO YOU WANT SIMPLE <S> OR COMPOUND <C>
    INTEREST?"
50 PRINT
    : PRINT "ENTER <S> OR <C>"
60 A$=INKEY$
    : IF A$="" THEN 60
70 IF A$="C" GOTO 100
80 IF A$="S" GOTO 780
90 GOTO 60
100 CLS
110 PRINT"  COMPOUND INTEREST CACULATOR"
120 FOR X=1 TO 460*2
    : NEXT
130 CLS
140 INPUT "ENTER PRINCIPAL";P
150 CLS
160 INPUT "ENTER RATE %";R
170 CLS
180 INPUT "ENTER NUMBER OF PERIODS PER YEAR";Y
190 CLS
200 INPUT "ENTER NUMBER OF YEARS";T
210 CLS
220 T1=T*Y
230 R=R/100
240 P1=P
    : C=0
    : I1=0
    : C1=0
    : C2=0
250 FOR X=1 TO T1
260 C=C+1
270 C2=C2+1
280 C1=C1+1
290 I=P*R/Y
300 P=P+I
310 I1=I1+I
320 IF L=1 GOTO 400
330 PRINT "PERIOD ";C1
340 PRINT USING "PRINCIPAL ########.##";P
350 PRINT USING "INTEREST  ########.##";I1
360 PRINT"*****************************"
370 IF C>2 THEN GOSUB  500
380 IF X=T1 GOTO 550
390 GOTO 460
400 IF C2>Y GOSUB 740
410 PRINT#-2,"PERIOD ";C1
420 PRINT#-2, USING "PRINCIPAL%
    %########.##";Z$,P
430 PRINT#-2, USING "TOTAL ACCUMULATED INTEREST
    #######.##";I1
440 PRINT#-2,"-------------------------------------"
450 IF X=T1 GOTO 730
460 NEXT X
470 CLS
    : GOTO 130
480 PRINT "OR <E> TO END"
490 END
500 IF X=T1 GOTO 550 ELSE 510
510 PRINT "PRESS <R> TO CONTINUE"
520 PRINT "PRESS <P> FOR PRINT OUT"
530 PRINT "OR PRESS <E> TO END"
540 GOTO 560
550 PRINT "INTEREST CALCULATIONS COMPLETE"
    : PRINT "PRESS <M> TO GO TO MENU"
    : GOTO 520
560 A$=INKEY$
    : IF A$="" THEN 560
570 IF A$="P" THEN 620
580 IF A$="E" THEN 490
590 IF A$="R" THEN C=0
    : CLS
    : RETURN
600 IF A$="M" GOTO 20
```

```
61Ø GOTO 56Ø
62Ø CLS
63Ø PRINT#-2, USING "BEGINNING PRINCIPAL
    ########.##";P1
64Ø PRINT#-2, USING "INTEREST RATE
    .###";R*1ØØ
65Ø PRINT#-2, "COMPOUNDED ";Y;"TIMES PER YEAR"
66Ø C3=1
67Ø PRINT#-2," "
68Ø PRINT#-2,"          YEAR";C3
69Ø PRINT#-2,"          ------"
7ØØ P=P1
    : L=1
71Ø C2=Ø
    : C1=Ø
72Ø GOTO 24Ø
73Ø PRINT#-2, "INTEREST COMPLETE"
    : GOTO 49Ø
74Ø C3=C3+1
75Ø PRINT#-2,"          YEAR";C3
76Ø PRINT#-2,"          ------"
77Ø C1=1
    : C2=1
    : RETURN
78Ø CLS
79Ø PRINT "SIMPLE INTEREST CALCULATOR"
8ØØ FOR X=1 TO 46Ø*2
    : NEXT
81Ø CLS
82Ø INPUT "ENTER PRINCIPAL";P
83Ø CLS
84Ø INPUT "ENTER RATE %";R
85Ø CLS
86Ø PRINT "IS THE TIME IN DAYS OR YEARS?"
87Ø PRINT
    : INPUT"      ENTER <D> OR <Y>";Y$
88Ø IF Y$="Y" THEN 99Ø
89Ø IF Y$="D" THEN 91Ø
9ØØ CLS
91Ø PRINT "IF NUMBER OF DAYS UNKNOWN PRESS  <D> OR
    ELSE HIT ANY OTHER KEY"
92Ø A$=INKEY$
    : IF A$="" THEN 92Ø
93Ø D$=A$
94Ø IF A$="D" THEN GOTO 117Ø
95Ø CLS
96Ø INPUT "ENTER NUMBER OF DAYS";T
97Ø T=T/365
98Ø GOTO 1ØØØ
99Ø INPUT "ENTER NUMBER OF YEARS";T
1ØØØ PRINT
1Ø1Ø R=R/1ØØ
1Ø2Ø I=P*R*T
1Ø3Ø P1=P+I
1Ø4Ø PRINT USING "BEGINNING PRINCIPAL
    #########.##";P1-I
    : PRINT
1Ø5Ø PRINT USING "EARNED INTEREST###########.##";I
1Ø6Ø PRINT
    : PRINT USING "ENDING PRINCIPAL
    ###########.##";P1
1Ø7Ø PRINT
    : PRINT "PRESS <S> TO ENTER NEW VALUES"
1Ø8Ø PRINT "PRESS <R> TO RETURN TO MENU"
1Ø9Ø PRINT "PRESS <P> FOR HARD COPY"
11ØØ PRINT "OR PRESS <E> TO END"
111Ø A$=INKEY$
    : IF A$="" THEN 111Ø
112Ø IF A$="S" THEN 81Ø
113Ø IF A$="R" THEN 2Ø
114Ø IF A$="P" THEN 153Ø
115Ø IF A$="E" THEN 49Ø
116Ø GOTO 111Ø
117Ø CLS
118Ø GOSUB 148Ø
119Ø INPUT "ENTER BEGINNING DATE <MM>, <DD>,
    <YYYY>";X,K,Z
12ØØ CLS
121Ø GOSUB 136Ø
    : A=N
122Ø GOSUB 134Ø
123Ø X1=X
    : K1=K
    : Z1=Z
    : B$=A$(W)
    : U1=U
124Ø INPUT "ENTER ENDING DATE <MM>, <DD>,
    <YYYY>";X,K,Z
125Ø GOSUB 136Ø
    : B=N
126Ø CLS
    : PRINT "BEGINNING DATE"
127Ø PRINT USING "###/###/#####%%%  %";X1,K1,Z1,Z$,B$
128Ø PRINT USING "DAY OF YEAR ####";U1
129Ø PRINT
    : PRINT "ENDING DATE"
13ØØ GOSUB 134Ø
131Ø T=ABS(A-B)
132Ø PRINT USING "DAYS BETWEEN=#######";T
133Ø GOTO 97Ø
134Ø PRINT USING "###/###/#####%%%  %";X,K,Z,Z$,A$(W)
135Ø PRINT USING "DAY OF YEAR ####";U
    : RETURN
136Ø V=Z
    : M=X-2
137Ø IF M<1 THEN M=M+12
    : V=V-1
138Ø C=INT(V/1ØØ)
    : D=V-C*1ØØ
139Ø F=INT((26*M-2)/1Ø) + K + D + INT(D/4) +
    INT(C/4)-2*C
14ØØ IF F<Ø THEN F=F+7
    : GOTO 14ØØ
141Ø W=F-INT(F/7)*7+27
142Ø N=V*365 + M*28 + K + INT((26*M-2)/1Ø) + INT(V/4)
    + INT(C/4) -C -31
143Ø U=M*28 + INT((M*26-2)/1Ø) + K + 29
    : E=365
144Ø G=(V/4ØØ=INT(V/4ØØ)) + ((V/4=INT(V/4))
    *(V/1ØØ<>INT(V/1ØØ)))
145Ø U=U+G
    : E=E+G
146Ø IF U>E THEN U=U-E
147Ø RETURN
148Ø A$(27)="SUN"
149Ø A$(28)="MON"
    : A$(29)="TUE"
15ØØ A$(3Ø)="WED"
    : A$(31)="THU"
151Ø A$(32)="FRI"
    : A$(33)="SAT"
152Ø RETURN
153Ø PRINT#-2,"SIMPLE INTEREST"
    : PRINT#-2," "
154Ø IF D$<>"D" THEN 162Ø
155Ø PRINT#-2, "BEGINNING DATE"
156Ø PRINT#-2, USING"###/###/#####%%%
    %";X1,K1,Z1,Z$,B$
157Ø PRINT#-2, USING "DAY OF YEAR ####";U1
    : PRINT#-2," "
158Ø PRINT#-2, "ENDING DATE"
159Ø PRINT#-2, USING "###/###/#####%%%
    %";X,K,Z,Z$,A$(W)
16ØØ PRINT#-2, USING "DAY OF YEAR ####";U
161Ø PRINT#-2," "
162Ø PRINT#-2, USING "TERM (DAYS)########";T*365
163Ø PRINT#-2, USING "INTEREST RATE%%.###";X$,R*1ØØ
164Ø PRINT#-2," "
    : PRINT#-2, USING "BEGINNING PRINCIPAL
    ##########.##";P1-I
```

```
1650 PRINT#-2, USING "INTEREST%
     %############.##";Z$,I
1660 PRINT#-2, USING "ENDING PRINCIPAL
     ############.##";P1
1670 CLS
     : GOTO 1070
```

Biorhythm for the Color Computer

Ed-Udom C. Javanaphet
4470 E. Sierra Madre #111
Fresno, CA 93726

This biorhythm program requires 16K extended BASIC, screen print software and a printer. The program not only puts your biorhythm on the screen, it also produces a printout. To produce a hardcopy, press (CR) (CR) when the chart has been completed on the screen.

If you do not have the older version of the screen print program (26-3021), or if you have a CGP-220 and would like to print the chart in color, you may want to add the new Hi-Res Screen Print Utilities package (26-3121) to your software collection. The Hi-Res Screen Print Utilities tape has a program on one side for Color printing with the CGP-220 and a program on the other side which will interface with any Radio Shack Dot Matrix Printer with Bit Image capabilities for monochrome printings: LPVIII and DMPs 100, 110, 120, 200, 400, and 420.

To use Mr. Javanaphet's Biorhythm program with the Hi-Res Screen Print Utilities package, lines 920 and 930 should be changed to read:

```
920 PRINT "ACTIVATE SCREEN PRINT":PRINT"PROGRAM"
930 PRINT"TYPE 'CONT' WHEN FINISHED'":STOP
```

Run the program, entering data as requested. After the chart has been printed on the screen, press (CR). The screen will display:

PRINTER ON 8/BS
?

Press (CR) again and the printer will print the data up to the chart. When the printer stops the screen display will read:

ACTIVATE SCREEN PRINT PROGRAM
TYPE 'CONT' WHEN FINISHED

Now activate the screen print program, following the instructions in the Hi-Res manual for the program you loaded to use with your printer. After the chart has been printed, you will need to press the left arrow key to delete the impression of the key used to activate the screen print program. Then just type "CONT" to complete the Biorhythm program.

```
10 REM *** BIORHYTHM PROGRAM BY ***
20 REM *** EK-UDOM C.JAVANAPHET ***
30 REM *** 5-28-1983 FRESNO, CA ***
40 CLS
     : PRINT@71,"*** BIORHYTHM ***"
50 A$="BIORHYTHM THEORY ORIGINATED IN EUROPE TOWARD
     THE END OF THE LAST CENTURY."
60 B$="EACH PERSON HAS A SET OF LIFE RHYTHMS OR LIFE
     CYCLES WHICH BEGINS THE DAY A PERSON IS BORN."
70 C$= "EACH OF THESE CYCLES TAKES THE FORM OF A SINE
     WAVE."
80 PRINT TAB(3)A$
     : PRINT TAB(3)B$
     : PRINT TAB(3)C$
90 PRINT
     : PRINT@459,"PRESS <CR>"
100 AN$=INKEY$
110 IF AN$=CHR$(13) THEN 120 ELSE 100
120 CLS
130 PI=3.141593
140 PRINT@71, "*** BIORHYTHM ***"
150 PRINT@133, "DATE FORMAT MM,DD,YYYY"
160 PRINT
     : PRINT
170 LINEINPUT "NAME : ";N$
180 PRINT
     : INPUT "TODAY DATE  ";TM,TD,TY
190 PRINT
     : INPUT "BIRTHDAY  ";BM,BD,BY
200 D=TD
     : M=TM
     : Y=TY
     : GOSUB 240
     : JT=J
210 D=BD
     : M=BM
     : Y=BY
     : GOSUB 240
     : JB=J
220 A=JT-JB
230 GOTO 280
240 IF M<=2 THEN X=0
     : YY=Y-1
     : GOTO 260
250 X=INT(.4*M+2.3)
     : YY=Y
260 J=INT((365*Y) + 31*(M-1)+D + (YY/4)-(YY/100) +
     (YY/400)-X)
270 RETURN
280 CLS
290 L=INT((32-LEN(N$))/2)
300 LL=LEN(N$)
310 PRINT@64+L,N$
     : PRINT@96+L, STRING$(LL,"*")
320 PRINT@128, "AGE : ";INT(A/365)
330 PRINT@160, "BIRTH : ";BM;"/";BD;"/";BY
340 PRINT@192, "OCCURRED ";A;" DAYS AGO"
350 D$="'P'= PHYSICAL CYCLE"
360 E$="'E'= EMOTIONAL CYCLE"
370 F$="'I'= INTELLECTUAL CYCLE"
380 PRINT@260,D$
     : PRINT@292,E$
     : PRINT@324,F$
390 PRINT
     : PRINT "BIORHYTHM START ON"
400 PRINTTM;"/";TD;"/";TY;" TILL NEXT MONTH"
410 PRINT@491, "PRESS <CR>"
420 AN$=INKEY$
430 IF AN$=CHR$(13) THEN 440 ELSE 420
440 B=23
     : GOSUB 480:P=V
450 B=28
     : GOSUB 480
     : E=V
460 B=33
     : GOSUB 480
     : I=V
470 GOTO 510
480 V=A
490 V=V/B
     : V=INT(V*100)/100
     : V$=STR$(V)
500 V$=RIGHT$(V$,2)
     : V=VAL(V$)
     : V=V/100
```

```
     : V=INT(V*B)
     : 'RETURN
510 PMODE 4,1
520 PCLS
530 SCREEN 1,1
540 LINE (127,5)-(127,185),PSET
550 LINE (7,95)-(247,95),PSET
560 FOR X=7 TO 247 STEP 20
570 PRESET(X,95)
580 NEXT X
590 FOR Y=5 TO 185 STEP 10
600 PRESET (127,Y)
610 NEXT Y
620 P$="U4R4D2L4"
     : B=23
     : GOSUB 650
630 P$="L4U3R4L4U2R4"
     : B=28
     : P=E
     : GOSUB 650
640 P$="L4R2U4R2L4"
     : B=33
     : P=I
     : GOSUB 650
     : GOTO 730
650 FOR I=8 TO 240 STEP 8
660 IF P=B+1 THEN P=1
670 T=INT(-(SIN((P/B)*2*PI))*10)
680 T=(T*6)+96
690 DRAW "BM"+STR$(I)+","+STR$(T)+"C1"+P$
700 P=P+1
710 NEXT I
720 RETURN
730 AN$=INKEY$
740 IF AN$=CHR$(13) THEN 760
750 GOTO 730
760 CLS
     : PRINT@136, "PRINTER ON 8/BS"
     : SOUND 150,2
     : INPUT H
770 PRINT#-2, CHR$(31)
     : PRINT#-2,TAB(12) "*** BIORHYTHM ***"
780 PRINT#-2, CHR$(30)
790 PRINT#-2
     : PRINT#-2
     : PRINT#-2
     : PRINT#-2,TAB(3);A$;B$;C$
800 PRINT#-2
     : PRINT#-2
     : PRINT#-2
810 PRINT#-2
     : PRINT#-2,TAB(5);D$;" (23 DAYS)"
820 PRINT#-2
     : PRINT#-2,TAB(5);E$;" (28 DAYS)"
830 PRINT#-2
     : PRINT#-2,TAB(5);F$;" (33 DAYS)"
840 PRINT#-2
     : PRINT#-2
850 PRINT#-2,TAB(20) "NAME : ";N$
860 PRINT#-2,TAB(20) "AGE  : ";INT(A/365)
870 PRINT#-2,TAB(20) "BORN ON ";BM;"/";BD;"/";BY
880 PRINT#-2,TAB(20) "OCCURRED ";A;" DAYS AGO"
890 PRINT#-2
     : PRINT#-2
900 PRINT#-2
     : PRINT#-2,TAB(20) "START ON ";TM;"/";TD;"/";TY;
     " TILL NEXT 30 DAYS"
910 PRINT#-2
     : PRINT#-2
920 DEFUSR0=15785
930 Y=USR(0)
940 PRINT#-2
     : PRINT#-2
     : PRINT#-2,TAB(50) "BY"
950 PRINT#-2,TAB(50) "TRS-80 COMPUTER"
960 END
```

Inventory Control

John Capazo
640 S. Monitor
West Point, NE 68788

I am enclosing an Inventory Control program I wrote for the appliance store where I work. The program is menu driven but I did not use the word menu in the program; I called it FILE. This program keeps track of the appliance type, model number, dealer cost, sell price, quantity on hand and the invoice number. I dimensioned my program to allow for only 100 different model numbers.

```
1 CLEAR 5000
3 DATA A$(N,R)
4 N= RND(100)
5 R=RND(7)
10 DIM A$(100,6)
15 CLS
     : PRINT @ 46,"FILE"
20 PRINT @ 71,"DO YOU WANT TO--"
30 PRINT @ 134,"(1) INPUT ITEMS"
40 PRINT @ 166,"(2) ADD IIEMS"
50 PRINT @ 198,"(3) LIST ITEMS"
60 PRINT @ 230,"(4) DELETE ITEMS"
70 PRINT @ 262,"(5) SAVE ITEMS ON TAPE"
80 PRINT @ 294,"(6) LOAD ITEMS FROM TAPE"
90 PRINT @ 326,"(1-6)
100 INPUT M
110 IF M<1 OR M>7 THEN 15
120 ON M GOTO 1000,2000,3000,4000,5000,6000
130 GOTO 15
1000 CLS
1010 PRINT @ 40,"--INPUT ITEMS--"
1020 PRINT "CHECK LIST FOR NUMBERS NOT USED.
     IF YOU USE ONE OF THOSE NUMBERS,
     YOU WILL ERASE THE INFORMATION ON THAT NUMBER."
1030 FOR X = 1 TO 460*6
     : NEXT X
1040 CLS
1050 PRINT "ENTER <0> UNDER ITEM NO. WHEN FINISHED"
1052 INPUT "ITEM NO.";N
1054 IF N=0 THEN 15
1056 IF N<0 OR N>100 THEN 1040
1060 INPUT "MODEL NO.";A$(N,1)
1070 INPUT "APPLIANCE";A$(N,2)
1080 INPUT "COST";A$(N,3)
1090 INPUT "SELL";A$(N,4)
1100 INPUT "QUANTITY ON HAND";A$(N,5)
1105 INPUT "INVOICE NO.";A$(N,6)
1110 GOTO 1040
2000 CLS
2005 DATA A$(N,R)
2010 PRINT @ 41,"--ADD ITEMS--"
2020 INPUT "ITEM NO.";N
2030 PRINT "MODEL NO: "A$(N,1)
2040 PRINT "APPLIANCE: "A$(N,2)
2050 PRINT "COST: "A$(N,3)
2060 PRINT "SELL: "A$(N,4)
2070 PRINT "QUANTITY ON HAND: "A$(N,5)
2080 Z=VAL(A$(N,5))
2090 INPUT "QUANTITY ADDED";Q
2095 Y=Z+Q
2100 A$(N,5)=STR$(Y)
2110 PRINT "NEW QUANTITY: "A$(N,5)
2120 PRINT "INVOICE NO: "A$(N,6)
2130 PRINT @ 417,"TYPE <SS> TO START OVER
     OR TYPE <FF> TO GO TO FILE."
2135 INPUT P$
2140 IF P$="SS" THEN 2000
2150 IF P$="FF" THEN 15
```

```
2160 GOTO 2130
3000 CLS
     :  PRINT @ 43,"--LIST--"
3005 DATA A$(N,R)
3010 PRINT "   DO YOU WANT TO--"
3020 PRINT " (1) SEE ENTIRE LIST"
3030 PRINT " (2) SEE ONLY ONE CATEGORY"
3040 INPUT " (1 OR 2)";Q
3050 IF Q<1 OR Q>2 THEN 3000
3060 ON Q GOTO 3100,3500
3070 GOTO 3000
3100 CLS
3110 DATA A$(N,R)
3115 INPUT "PRESS <ENTER> TO START LIST";B$
3120 FOR N = 1 TO 100
3125 CLS
     : PRINT N
3130 PRINT "MODEL NO: "A$(N,1)
3135 IF A$(N,1) ="" THEN PRINT @74,"OPEN NUMBER"
3140 PRINT "APPLIANCE: "A$(N,2)
3150 PRINT "COST: "A$(N,3)
3160 PRINT "SELL: "A$(N,4)
3170 PRINT "QUANTITY ON HAND: "A$(N,5)
3180 PRINT "INVOICE NO: "A$(N,6)
3185 FOR X=1 TO 1000
     : NEXT X
3190 NEXT N
3200 PRINT @ 417,"TYPE <SS> TO START OVER
     OR TYPE <FF> TO GO TO FILE"
3205 INPUT P$
3210 IF P$="SS" THEN 3000
3220 IF P$="FF" THEN 15
3500 CLS
3510 DATA A$(N,R)
3520 PRINT " WHICH APPLIANCE DO YOU WANT TO  LIST?"
3530 PRINT "-REFRIGERATORS"
3535 PRINT "-FREEZERS"
3540 PRINT "-DISHWASHERS"
3550 PRINT "-CMOS"
3560 PRINT "-CMO-HOODS"
3570 PRINT "-RANGE-CMOS"
3580 PRINT "-WASHERS"
3590 PRINT "-DRYERS"
3591 PRINT "-TVS"
3592 PRINT "-CONSOLE TVS"
3593 PRINT "-AIR CONDITIONERS"
3600 INPUT "APPLIANCE";C$
3695 DATA A$(N,R)
3700 DATA A$(N,R)
3710 CLS
3720 FOR N=1 TO 100
3730 IF C$=A$(N,2) THEN 3750
3735 CLS
3740 GOTO 3830
3750 CLS
     : PRINT A$(N,2)
3760 PRINT N
3770 PRINT "MODEL NO: "A$(N,1)
3780 PRINT "COST: "A$(N,3)
3790 PRINT "SELL: "A$(N,4)
3800 PRINT "QUANTITY ON HAND: "A$(N,5)
3810 PRINT "INVOICE NO: "A$(N,6)
3820 FOR X = 1 TO 1500
     : NEXT X
3830 NEXT N
3840 GOTO 3200
4000 CLS
4005 DATA A$(N,R)
4010 PRINT @ 40,"--DELETE ITEMS--"
4020 INPUT "ITEM NO";N
4030 PRINT "MODEL NO: "A$(N,1)
4040 PRINT "APPLIANCE: "A$(N,2)
4050 PRINT "COST: "A$(N,3)
4055 IF VAL(A$(N,3)) = 0 THEN 4090
4060 INPUT "SALE PRICE";S
4070 C=VAL(A$(N,3))
4080 T=S-C
4081 P=T/C
4085 PRINT "PROFIT MARGIN: "P
4090 PRINT "QUANTITY ON HAND: "A$(N,5)
4100 Z=VAL(A$(N,5))
4110 INPUT "NUMBER OF ITEMS SOLD";I
4120 Y=Z-I
4130 A$(N,5)=STR$(Y)
4135 PRINT "QUANTITY LEFT: "A$(N,5)
4140 PRINT "INVOICE NO: "A$(N,6)
4150 PRINT @ 417,"TYPE <SS> TO START OVER
     OR TYPE <FF> TO GO TO FILE"
4155 INPUT P$
4160 IF P$ ="SS" THEN 4000
4170 IF P$ ="FF" THEN 15
4180 GOTO 4150
5000 CLS
5001 DATA A$(N,R)
5004 PRINT "REWIND TAPE AND PRESS PLAY ONLY"
5005 INPUT "PRESS <ENTER> WHEN READY";E$
5006 CLS
     : SKIPF "INVENTORY"
5020 PRINT @ 294,"PRESS PLAY AND RECORD"
5030 PRINT @ 388,"PRESS <ENTER> WHEN READY"
5040 INPUT R$
5045 CLS
     : PRINT "COMPUTER IS PROCESSING"
5050 OPEN "O",#-1,"LIST"
5060 FOR N=1 TO 100
5070 PRINT #-1, A$(N,1)
5080 PRINT #-1, A$(N,2)
5090 PRINT #-1, A$(N,3)
5100 PRINT #-1, A$(N,4)
5110 PRINT #-1, A$(N,5)
5120 PRINT #-1, A$(N,6)
5130 NEXT N
5140 CLOSE #-1
     :  GOTO 15
6000 CLS
     : PRINT "REWIND THE RECORDER AND PRESS PLAY"
6010 INPUT "PRESS <ENTER> WHEN READY";R$
6015 CLS
     : PRINT "COMPUTER IS PROCESSING"
6020 OPEN "I",#-1,"LIST"
6025 IF EOF (-1) THEN 6100
6029 FOR N = 1 TO 100
6030 INPUT #-1, A$(N,1)
6040 INPUT #-1, A$(N,2)
6050 INPUT #-1, A$(N,3)
6060 INPUT #-1, A$(N,4)
6070 INPUT #-1, A$(N,5)
6080 INPUT #-1, A$(N,6)
6090 NEXT N
6100 CLOSE #-1
     : GOTO 15
```

Learning at Home with Color Math

by Kathy Frasca Priest

The new Radio Shack Color Math Program is designed so that your child can practice, in a home environment, the basic math concepts he or she is learning at school. It consists of four topics: Addition, Subtraction, Multiplication and Division, and offers placement, skill building lessons and testing options in each. In all four subject areas, your child can be placed by the computer at his or her appropriate skill level, take lessons at that level followed by reports on his or her performance, and take a test on any lesson. Color Math is designed for use on the TRS-80 16K Color Computer tape system with any color television, and is suitable for children in grades one through eight.

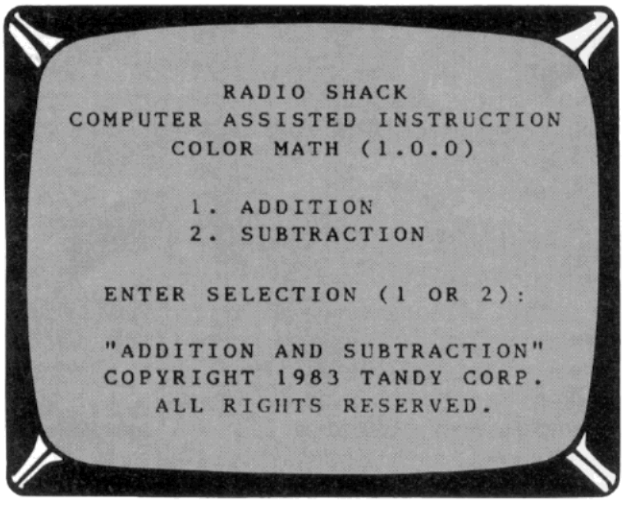
With the Color Math Program, your child can:

- automatically be placed by the computer in the lesson which most closely corresponds to his or her present skill level in each of the four math topics;
- use the skill building lessons in each subject area to increase his or her math skills, with automatic promotion as each lesson is mastered; and
- take a test on any lesson in the program, to evaluate his or her proficiency at that level.

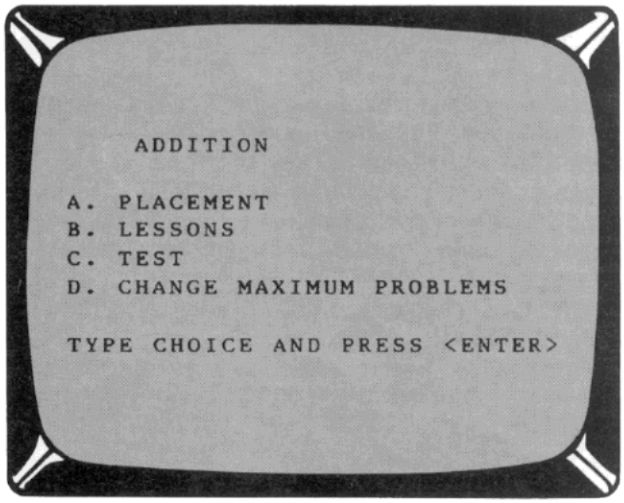
USING THE COLOR MATH PROGRAM

One Color Math program, containing two topics, is stored on each side of the cassette tape provided. Addition and Subtraction lessons are on Side One, while Side Two contains Multiplication and Division. For this demonstration, we'll use Side One.

Once the Color Math Program is loaded, the first screen of the program will appear:



This is the Topic Option Screen: it lets you choose between the two topics offered on this side of the cassette tape. Let's choose Addition. After the number [1] is typed in, the computer will ask you for the number of problems you want your child to do in an Addition Lesson. Type [2] [0] (or whatever number you want) and press [ENTER]. The next screen will appear:



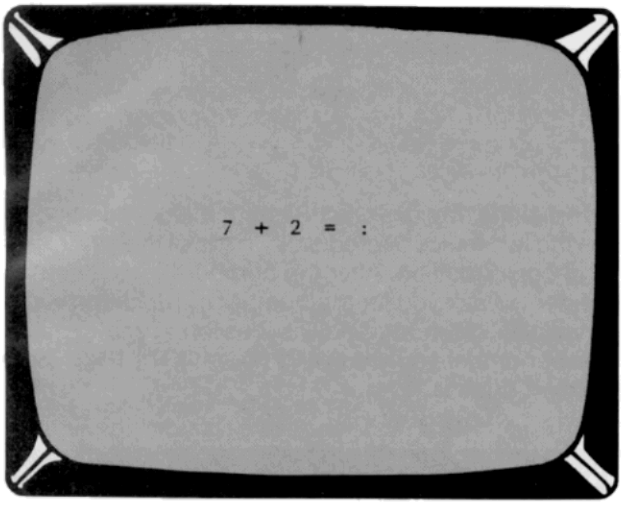
This display is called the Option List, and it gives you four choices. Choose:

- [A] to have the computer find the appropriate starting lesson for your child (that is, to "place" your child in the program),
- [B] to let your child take skill building lessons,
- [C] to have your child take a test (on any lesson), or
- [D] to change the maximum number of problems you entered earlier.

The first thing you should do when starting your child in any of the four subject areas for the first time is to use the Placement option [A]. The purpose of Placement is to find the lesson where your child should start in the Addition, Subtraction, Multiplication or Division program. During Placement, the computer will automatically evaluate your child's performance in each of the four subjects and place him or her in the appropriate lesson. The computer will generate as many problems as necessary to determine where your child should be placed. Once the computer has found the appropriate lesson, you'll see a report screen telling you the results.

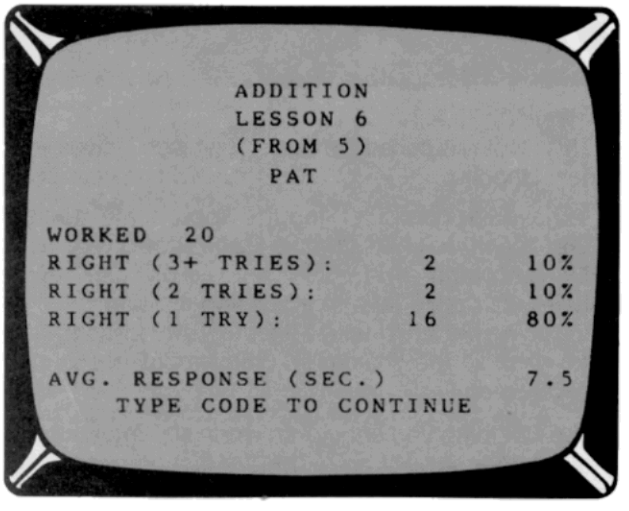
The skill building lessons in Color Math will provide your child with intensive practice in math skills. Simply choose [B] at the Option List, type in the number of problems you would

like your child to do, your child's name, and the number of the lesson in which he or she was just placed. The first problem from that lesson will immediately appear:



The tiny blinking squares to the right of the equals sign show where your answer will appear on the screen. To solve a problem, press the number key(s) of your answer. If your answer is correct, you will hear a short tone or beep, and a "Happy Face" will appear on the screen. (When your child reaches the fourth-grade level of lessons, the "Happy Face" is replaced by a "CORRECT" message.)

When your child has completed the specified number of problems, a report on his or her performance will appear, similar to the following example:



The report tells you how many problems were worked, how many were answered correctly on 1, 2, and 3 or more (3+) tries, the percent correct on 1, 2, and 3+ tries, and the average response time in seconds. For example, the report above shows that Pat worked 20 problems in Addition Lesson 5, getting 16 correct on the first try (80%), 2 correct on the second try (10%), and 2 correct after 3 or more tries (10%). The average time between keystrokes was 7.5 seconds. Since Pat did so well, the screen shows a promotion to Lesson 6. The top three lines on the screen tell you that Pat has been moved up to the next lesson.

If your child does not do very well on the lesson, the computer may show a demotion to the previous lesson. This indicates that the lesson just taken is too difficult at this time and that your child needs practice in an easier lesson.

As you can see, Color Math lets your child work through the lessons at his or her own pace. When a lesson has been mastered, the child is promoted to the next higher one. If a lesson proves to be too difficult, the child is moved back to the previous one for more practice. Sometimes the child will do well enough to avoid being demoted, but not well enough to be promoted. In this case, the child will remain in the same lesson for more practice, and no promotion or demotion will occur.

OTHER FEATURES OF THE PROGRAM

In advanced Addition and Subtraction lessons, carrying or borrowing occurs in almost every problem. Your child will get help from the computer when a problem requires carrying or borrowing. The flashing squares (the "cursor") will appear at the position where the carried or borrowed digit should be typed in. In advanced Multiplication lessons, your child is aided by the "operand flash," in which the digits to be multiplied flash on the screen.

A performance report will automatically appear on the screen when the specified number of problems has been completed. However, you do not have to wait for the end of the lesson; you can press (R) and then (T) at any time during a lesson or test and a report will appear with information on the problems worked so far.

The lessons in all four Color Math topics have been designed so that a minimum number of problems must be worked before the computer will check to see if a child should be promoted, demoted or remain at the same level. The percent score required for each of these three conditions also varies from lesson to lesson. The Mastery Levels lists included in the package give this information for each lesson.

You can give your child a timed lesson by typing a zero in response to the "Maximum Problems?" prompt. The computer will present an unlimited number of problems until you end the session.

You can give your child a test on any lesson in Color Math. Simply choose TEST at the Option List and type in your child's name, the lesson number, and the number of problems. During a test, your child has only one chance to answer each problem, and has no indication of whether the answers given are right or wrong. At the end of the specified number of problems, a report will appear with the score. If you want to give your child a timed test, enter zero for the number of problems, and type (R) (T) to get a report when the time is up.

MORE INFORMATION ABOUT COLOR MATH

The Color Math manual includes:

- A Record Sheet onto which you should record all scores from lessons and tests. This will give you a complete record of your child's achievement and will keep track of his or h progress through the Color Math lessons.
- A Sample Problems Sheet for each of the four topics. These sheets show at least one example of the type of problem you will see in each lesson.

- A Lesson Content Summary for each of the four topics. These give a brief description of the problem types in each subject area, explain the cursor cueing feature, tell how to correct errors, and contain a glossary of terms.
  A Cross-Reference which correlates individual Color Math lessons with specific pages in seven textbook series used in schools today. Using this reference, you can have your child work on the Color Math lessons which reinforce the textbook material he or she is currently studying in school.

In order for your child to derive maximum benefit from this program, it is recommended that sessions at the computer are limited to a maximum of 10 minutes for children in grades 1 through 3, and 15 minutes for those in grades 4 through 8. There should be no more than two sessions per day. Limiting the time insures that your child will remain interested and look forward to each session. Although Color Math is entertaining and designed to motivate your child, it is not a game. Each program provides intensive exercise in math concepts, and should provide your child with many hours of challenging and effective practice.

The Radio Shack Color Math Program (Catalog #26-3201) will be available soon through your local Radio Shack store or Computer Center. Or contact the Radio Shack Regional Educational Coordinator in your area. Radio Shack has a network of 25 Regional Educational Coordinators nationwide to help schools and districts with their computing needs.

Easter Bunny

Vergil English
4559 Campus Ave.
San Diego, CA 92116
CompuServe ID 71675,1536

Happy Easter from the talking bunny. This program was written for a 16K color computer.

```
5 BY V. ENGLISH
    : 4559 CAMPUS AVE
    : SAN DIEGO, CA 92116
20 CLEAR 2000
30 PMODE 3, 1
40 PCLS
50 SCREEN 1, 1
100 CIRCLE(100,130),35,,1,.10,.66
200 CIRCLE(156,130),35,,1,0,.40
300 CIRCLE(128,120),55,,1,.54,.98
310 CIRCLE(156,130),35,,1,.80,0
400 DRAW "BM160,80U15R20F55D30H60L5" 'EARS
410 DRAW "BM90,80H50L10G10F60"
420 CIRCLE(100,95),10" 'EYES
430 CIRCLE(150,95),10
440 CIRCLE(128,140),13 'NOSE
450 LINE(110,160)-(140,160),PSET 'MOUTH
455 PAINT(10,20),3,4
460 FOR X = 1 TO 2
470 FOR Y = 1 TO 15 STEP 3
480 PAINT(128,153),3,4
485 FOR T = 1 TO 30
    : NEXT T
490 PAINT(128,158),2,4
495 SOUND 247,1
    : SOUND 248,1
500 NEXT Y
510 IF X=1 THEN GOSUB 1000
520 NEXT X
530 GOSUB 2000
550 LINE(90,95)-(110,97),PSET 'BLINKING
560 LINE(90,97)-(110,97),PSET
570 LINE(90,100)-(110,100),PSET
575 FOR T = 1 TO 130
    : NEXT T
580 LINE(90,100)-(110,100),PRESET
590 LINE(90,97)-(110,97),PRESET
600 LINE(90,95)-(110,95),PRESET
650 LINE(100,150)-(110,160),PSET
660 LINE(140,160)-(150,150),PSET
670 CIRCLE(100,100),3
680 CIRCLE(150,100),3
700 FOR X = 1 TO 6
710 LINE(118,135)-(118,145),PSET
720 LINE(137,135)-(137,145),PSET
730 FOR T = 1 TO 10
    : NEXT T
740 LINE(118,135)-(118,145),PRESET
750 LINE(137,135)-(137,145),PRESET
760 LINE(120,135)-(120,145),PSET
770 LINE(138,135)-(138,145),PSET
780 FOR T = 1 TO 10
    : NEXT T
790 LINE(120,135)-(120,145),PRESET
800 LINE(138,135)-(138,145),PRESET
810 NEXT X
820 CIRCLE(128,140),13
900 GOTO 900
1000 DRAW "BM40,170D10BR10U5NL10U5BR5" 'H
1010 DRAW "BM55,170ND10R10D5NL10D5" 'A
1020 DRAW "BM70,170ND10R10D5L10" 'P
1030 DRAW "BM85,170ND10R10D5L10" 'P
1050 DRAW "BM100,170F5ND5E5" 'Y
1060 RETURN
2000 DRAW "BM130,170NR10D5NR10D5R10" 'E
2010 DRAW "BM145,170ND10R10D5NL10D5" 'A
2020 DRAW "BM160,170NR10D5R10D5L10" 'S
2030 DRAW "BM175,170R5ND10R5" 'T
2040 DRAW "BM190,170NR10D5NR10D5R10" 'E
2050 DRAW "BM205,170ND10R10D5L10F11" 'R
2060 RETURN
```

430 Radio Shack Computer Centers

ALABAMA
BIRMINGHAM 2428 Green Springs Hwy., 9233 Parkway East
HUNTSVILLE 1400 N. Memorial Pkwy.
MOBILE 405 Bel-Air Blvd.
MONTGOMERY #24 Union Square S/C

ARIZONA
PHOENIX 10233 Metro Pkwy. E., 4301 N. 7th Ave.
SCOTTSDALE 2525 N. Scottsdale Rd.
TEMPE 83 E. Broadway
TUCSON 5622 E. Broadway, Campbell Plaza, 2830 N Campbell Ave.

ARKANSAS
LITTLE ROCK Town & Country S/C, University & Asher

CALIFORNIA
ANAHEIM 509 Katella
BAKERSFIELD 2018 Chester Ave.
BERKELEY 1922 Grove St.
BEVERLY HILLS 8500 Wilshire Blvd.
BREA Imperial Shopping Center, 391 South State College
CANOGA PARK 8371 Topanga Canyon
CARMICHAEL 6305 Fairoaks Blvd
CHICO 1834 Mangrove Ave.
CHULA VISTA 1201 3rd Ave.
CITRUS HEIGHTS 7405 Green Back Ln. at San Juan
DOWNEY 8031 Florence Ave.
EL TORO El Toro Rd. at San Diego Freeway
ESCONDIDO 347 W. Mission Ave.
FREMONT Fremont Hub, 39114 Fremont
FRESNO Princeton S/C, 2721 N. Blackstone Ave.
GARDEN GROVE 12821 Knott St.
GLENDALE 236 N. Brand Blvd.
HAYWARD 24784 Hesperian Blvd.
HOLLYWOOD 6922 Hollywood Blvd.
IRVINE Redhill at Main St.
LAKEWOOD 5830 Lakewood Blvd.
LA MESA 5346 Jackson Dr.
LONG BEACH 2119 Bellflower Blvd.
LOS ANGELES 740 S. Olive St., 5240 Century Blvd. (Airport area)
MODESTO 221 McHenry Ave.
MONTCLAIR 5237 Arrow Hwy.
MONTEREY 484 Washington St.
MOUNTAIN VIEW 1933 El Camino Real W.
OAKLAND 1733 Broadway, (415) 763-3183
PASADENA 575 S. Lake Ave.
PLEASANT HILL 508 G. Contra Costa Blvd.
RIVERSIDE 3844 La Sierra Ave.
SACRAMENTO 4749 J. St.
SAN BERNARDINO 764 Inland Center Dr.
SAN DIEGO 3062 Clairemont Dr., 3902 El Cajon Blvd.
SAN FRANCISCO One Market Place, 2920 Geary Blvd.
SAN JOSE 1228 S. Bascom Ave.
SAN MATEO 3180 Campus Dr.
SANTA ANA 2320 S. Fairview St. (at Warner)
SANTA BARBARA 4141 State St. A-1
SANTA FE SPRINGS 14138 East Firestone Blvd.
SANTA MONICA 511 Wilshire Blvd.,
SANTA ROSA 823 4th St.
SHERMAN OAKS 14936 Ventura Blvd.
STOCKTON College Sq. S/C, 963 West March Lane
TARZANA 18545 Ventura Blvd.,
TORRANCE 3840 Sepulveda at Hawthorne
VENTURA 4005 E. Main St.
WEST COVINA 2516 E. Workman St.

COLORADO
AURORA Hoffman Heights Shop. Ctr.
BOULDER Arapahoe Plaza, 3550 Arapahoe
COLORADO SPRINGS 4341 N. Academy
DENVER 8000 E. Quincy, Green Briar Plaza, 7075 Pecos
LAKEWOOD 2099 Wadsworth Blvd.

CONNECTICUT
EAST HAVEN 51 Frontage Rd.
FAIRFIELD 1196 Kings Hwy. & Rt. 1
HARTFORD The Richardson Bldg., 942 Main St.
MANCHESTER 228 Spencer St.
NORWALK Rt. 7-345 Main Ave.
ORANGE Edwards Food Warehouse Shop. Ctr., 538 Boston Post Rd.
WATERBURY 105 Bank St.
WATERFORD 122 Boston Port Rd.
WEST HARTFORD 39 S. Main St.

DELAWARE
DOVER Edgehill Shop. Ctr., Rt. #13
WILMINGTON 3847 Kirkwood Hwy.

DISTRICT OF COLUMBIA
N.W. WASHINGTON 1800 M St., 15th & L St., Van Ness Station, 4250 Connecticut Ave.

FLORIDA
ALTAMONTE SPRINGS 766 B. East Altamonte Dr.
BOCA RATON 1662 N. Federal Hwy.
CLEARWATER 2460-D US 19 North
DAYTONA BEACH Volusia Plaza
FT. LAUDERDALE 4368 N. Federal Hwy.
FORT MYERS Edison Mall
FORT PIERCE Center West Shopping Center, U.S. Hwy. 1
GAINESVILLE 3315 Archer Rd.
HOLLYWOOD 429 S. State Rd. #7
JACKSONVILLE 8252 Arlington Exprwy., Roosevelt Mall, Roosevelt Blvd.
LAKELAND 2810 S. Florida Ave.
LAUDERDALE LAKES 4317-25 N. State Rd. 7
MERRITT ISLAND 700 E. Merritt Island Causeway
MIAMI 9459 S. Dixie Hwy., 1601 Biscayne Blvd., 15 SE. 2nd Ave., 20761 S. Dixie Hwy.
N. MIAMI BEACH The Promenade, 1777 N.E. 163rd St.
ORLANDO 1238 E. Colonial Dr.
PENSACOLA Eastgate Shopping Center, 6925 N. 9th Ave.
ST. PETERSBURG 3451 66th St. N.
SARASOTA 5251 S. Tamiami Tr. (Hwy.41)
TALLAHASSEE 2529 S. Adams
TAMPA 4555 W. Kennedy, 1825 E. Fowler Ave.
W. PALM BEACH 2271-A Palm Beach Lakes Blvd

GEORGIA
ATLANTA 2106 Henderson Mill, 49 W. Paces Ferry, Akers Mill S/C, 2937 Cobb Parkway NW., 113 Peachtree St.
AUGUSTA 3435 Wrightsboro Rd.
COLLEGE PARK 5309 Old National Hwy.
COLUMBUS Columbus Square Mall, 3050 Macon Rd.
DOUGLASVILLE 5697 Buford Hwy.
MACON 1467 Eisenhower Pkwy.
SAVANNAH Chatham Plaza, 7805 Abercorn St.

HAWAII
HONOLULU Pacific Trade Center-Ground Floor, 190 South King St.

IDAHO
BOISE 691 S. Capitol Blvd.

ILLINOIS
AURORA 890 North Lake St.
BLOOMINGTON 401 N. Veterans Parkway (Eastland Commons)
CHICAGO 4355 S. Archer Ave., CNA Plaza, 309 S. Wabash, 72 West Adams, 524 N. Michigan Ave., 101 W. Washington St. at Clark
DEERFIELD Deerbrook Mall
ELMWOOD PARK 7212 W. Grand Ave.
FAIRVIEW HEIGHTS #4 Market Place
HOMEWOOD/GLENWOOD 329 Glenwood Lansing, 18230 S. Halsted at 183rd
JOLIET 2415 W. Jefferson St.
LaGRANGE One S. LaGrange Rd.
LIBERTYVILLE 1350 S. Milwaukee Ave.
LOMBARD 4 Yorktown Center
MOLINE 4401 44th Ave.
NILES 8349 Golf Rd.
OAK LAWN 4815 W. 95th St.
PEORIA 4125 N. Sheridan Rd.
ROCKFORD North Town S/C, 3600 N. Main St.
SCHAUMBURG 651 Mall Dr.
SPRINGFIELD Sherwood Plaza, 2482 Wabash

INDIANA
EVANSVILLE 431 Diamond Ave.
FT. WAYNE 747 Northcrest S/C
GRIFFITH 208 W. Ridge Rd.
HOBART Save-More Plaza, Rtes. #6 & 51
INDIANAPOLIS 6242 E. 82nd St., Castleton Plz., Speedway Plaza, 6129 B Crawfordsville, 10013 E. Washington St., Two W. Washington St.
SOUTH BEND 1827 South Bend Ave.
TERRE HAUTE 3460 U.S. Hwy. 41 S.

IOWA
CEDAR RAPIDS 111 First Ave., S.E. (Downtown)
DAVENPORT 616 E. Kimberly Rd.
DES MOINES 7660 Hickman Rd., Sherwood Forest S/M

KANSAS
OVERLAND PARK 8619 W. 95th
TOPEKA White Lakes Plaza, West Tower, 3715 Plaza Dr.
WICHITA 2732 Blvd. Plaza S/C

KENTUCKY
FLORENCE 7727 Mall Rd.
LEXINGTON 2909 Richmond Rd.
LOUISVILLE Louisville Galleria, 4133 Shelbyville Rd.

LOUISIANA
ALEXANDRIA 1213 Texas Ave.
BATON ROUGE 7007 Florida Blvd.
GRETNA 400 La Palco Blvd.
HOUMA 2348 W. Park Ave. (Hwy. 24)
LAFAYETTE University Square at Congress Blvd.
METAIRIE 3750 Veterans Hwy.
NEW ORLEANS 327 St. Charles Ave.
SHREVEPORT 1545 Line Ave.

MAINE
BANGOR Maine Square

MARYLAND
BALTIMORE 7942 Belair Rd., Putty Hill Plaza, 115 N. Charles St. at Lexington
BETHESDA 7900 Wisconsin Ave.
CATONSVILLE One Mile West S/C, 6600 B Balt. Nat'l. Pike
FREDERICK Shoppers World, Rt. 40W
NEW CARROLLTON-LANHAM 7949 Annapolis Rd.
PASADENA 8120 Ritchie Hwy.
ROCKVILLE Congressional Plaza, 1673 Rockville Pike
SALISBURY Shoppers World S/C, Rt. 50
TEMPLE HILLS 4520 St. Barnabas Rd.
TOWSON-LUTHERVILLE Yorktown S/C York Rd. at Ridgley Rd.

MASSACHUSETTS
BOSTON 730 Commonwealth Ave., 111 Summer St., 372 Boylston
BOSTON 372 Boylston
BRAINTREE South Shore Plaza, 250 Granite St.
BROCKTON 675 Belmont
BURLINGTON Crossroads Plaza, Rt. 3 S.
CAMBRIDGE Harvard Square, 28 Boylston St.
CHESTNUT HILL 200 Boylston St.
FALL RIVER 90 William S. Canning Blvd
NATICK 1400 Worcester Rd.
SAUGUS 343 Broadway
SPRINGFIELD 1985 Main St., Northgate Plz.
WORCESTER Lincoln Plaza

MICHIGAN
ANN ARBOR 2515 Jackson Rd.
BIRMINGHAM 3620 W. Maple Rd.
DEARBORN Westborn Shop. Ctr., 23161 Michigan Ave.
DETROIT DWNTN 1559 Woodward Ave.
FLINT G3298 Miller Rd., Yorkshire Plaza
GRAND RAPIDS 3142 28th St. SE.
KALAMAZOO 25 Kalamazoo Center
LANSING 2519 S. Cedar St.
LIVONIA 33470 W. 7 Mile Rd.
PLAINFIELD North Kent Mall,
PONTIAC North Oak Plaza-Pontiac Mall, 2436 Elizabeth Lake Rd.
ROSEVILLE 31873 Gratiot Ave.
SOUTHFIELD 17651 West 12 Mile Rd.,
TROY Oakland Plaza, 322 John R. Rd.
WARREN 29038 Van Dyke Ave.

MINNESOTA
BLOOMINGTON 10566 France Ave. S.
FRIDLEY 7974-76 University Ave. North
GOLDEN VALLEY Golden Valley S/C, 8016 Olson Memorial Hwy.
MINNEAPOLIS 830 Marquette Ave.
ST. PAUL 6th & Wabasha

MISSISSIPPI
GULFPORT 516A Courthouse Rd.
JACKSON 979 Ellis Ave.

MISSOURI
DES-PERES 11960 Manchester Rd.
FLORISSANT 47 Florissant Oaks S/C
INDEPENDENCE 1325 S. Noland Rd.
KANSAS CITY 4025 N. Oak Trafficway
ST. ANN 10472 St. Charles Rock Rd.
ST. LOUIS 500 No. Broadway (Commerce Bank Bldg., Downtown)
SPRINGFIELD 2684 S. Glenstone

NEBRASKA
LINCOLN 4601 "O" St.
OMAHA 3006 Dodge St., 1318 72nd St. at Pacific

NEVADA
LAS VEGAS Commercial Center, 953 E. Sahara #31-B
RENO 3328 Kietzke Lane

NEW HAMPSHIRE
MANCHESTER Hampshire Plaza, 1000 Elm St.
NASHUA 429 Amherst St., Rt. 101A

NEW JERSEY
BRIDGEWATER 1472 U.S. Highway 22 East
E. BRUNSWICK 595 A Rt. 18
E. HANOVER Rt. 10, Hanover Plaza
LAWRENCEVILLE Rt. 1 & Texas Ave.
NEWARK 595 Broad
NORTHFIELD 322-24 Tilton Rd.
PARAMUS 175 Rt. 17 S.
POMPTON PLAINS Plains Plaza
SPRINGFIELD Rt. #22 Center Isle
TOMS RIVER 700 Rt. 37 West
VOORHEES 35 Eagle Plaza

NEW MEXICO
ALBUQUERQUE 2108 San Mateo NE

NEW YORK
ALBANY Shoppers Pk., Wolf Rd.
BAYSHORE 1751 Sunrise Hwy.
BETHPAGE 422 N. Wantagh Ave.
BROOKLYN 531 86th St.
BUFFALO 839 Niagara Falls Blvd.
FRESH MEADOWS 187-12 Horace Harding Exp.
GARDEN CITY 960 Franklin Ave.
JOHNSON CITY Giant Shopping Center, Harry L. Drive
KINGSTON Kings Mall, Rt. 9W
MANHASSET 1550 Northern Blvd.
MELVILLE TSS Mall, Rt. 110
NEWBURGH Zayre Plaza, Rt. #17K
NEW ROCHELLE 211 North Ave.
NEW YORK 385 Fifth Ave., 139 E. 42nd St., 19 W. 23rd St., 347 Madison Ave., 270 Park Ave. South, 1282 Broadway, 9 Broadway
NIAGARA FALLS Pine Plaza, 8351 Niagara Falls Blvd.
REGO PARK 97-77 Queens Blvd.
ROCHESTER 3000 Winton Rd.
SCARSDALE 365 Central Park Ave.
SCHENECTADY Woodlawn Plaza
SPRING VALLEY White House Center, 88 W. Rt. 59
STATEN ISLAND 2409 Richmond Ave.
SYRACUSE 2544 Erie Blvd., Hotel Syracuse, 510 S. Warren St.
UTICA Riverside Mall
VALLEY STREAM Green Acres Shop. Ctr.
YONKERS Cross Country Shop. Ctr.

NORTH CAROLINA
ASHEVILLE K-Mart Shopping Center, Tunnel Rd.
CHARLOTTE 3732 Independence Blvd., Tyvola Mall, 5401 South Blvd.
DURHAM South Square Mall
FAYETTEVILLE Eutaw Shopping Center, 815 Elm St., Fayetteville Street Mall
GREENSBORO 3718 High Point Rd.
RALEIGH Townridge Sq., Hwy. 70 W.
WILMINGTON Independence Mall
WINSTON-SALEM 629 Peters Creek Pkwy.

OHIO
AKRON Fairlawn Plaza, 2727 W. Market St.
BEDFORD HEIGHTS 5217 Northfield Rd.
CANTON 5248 Dressler Rd. NW., Mellet Plaza, 3826 W. Tuscarawas
CENTERVILLE 2026 Miamisburg-Centerville Rd.
CINCINNATI 9725 Montgomery, 16-18 Convention Way (on Skywalk)
CLEVELAND 419 Euclid (Dwntwn), 27561 Euclid Ave.
COLUMBUS 862 S. Hamilton, Great Eastern S/C, The Patio Shop. Ctr., 4661 Karl Rd., 400 N. High St.
DAYTON Northwest Plaza, 3279 West Siebenthaler
ELYRIA 286 Midway Blvd.
FAIRFIELD 7255 Dixie Hwy. (1/4 Mi. North of I-275)
NORTH OLMSTED Great Northern S/C
PARMA 7551 W. Ridgewood Dr.
TOLEDO 5844 W. Central Ave., Brownstone Plaza, 1724 S. Reynolds Rd.
YOUNGSTOWN Union Square Plaza, 2543 Belmont Ave.

OKLAHOMA
OKLAHOMA CITY 4732 SE 29th St., Springdale S/C, 4469 NW 50th, 1101 SW 59th St.
TULSA 7218 & 7220 E. 41st St.

OREGON
EUGENE 390 Coburg Rd.
PORTLAND 7463 SW Barbur Blvd., 9131 SE Powell, 3rd and Washington Sts. (Downtown)
SALEM Salem Plaza, 403 Center

PENNSYLVANIA
ALLENTOWN Crest Plaza S/C, Cedar Crest Blvd. US 22
BALA CYNWYD 67 E. City Line Ave.
EASTON 25th St. Shopping Center
ELKINS PARK Elkins Park Square, 8080 Old York Rd.
ERIE 5755 Peach St.
HARRISBURG Union Deposit Mall, Union Deposit Rd. #17
LANCASTER Park City Plaza, US 30
MONROEVILLE 3828 Wm. Penn. Hwy.
MONTGOMERYVILLE Airport Sq., Rt. 309
PHILADELPHIA 7542 Castor Ave., 1002 Chestnut St., 1801 Market St., 10 Penn Center
PITTSBURGH 5775 Baptist Rd., Hills Plaza, 303 Smithfield St., 4643 Baum Blvd., 4768 McKnight Rd.
SCRANTON 206 Meadow Ave.
WILKES BARRE, 23 W. Market St.
WYOMISSING Berkshire Mall West, 1101 Woodland Rd.
YORK York County Shopping Center

PUERTO RICO
HATO REY 243 Franklin D. Roosevelt Ave.

RHODE ISLAND
E. PROVIDENCE 850 Waterman Ave.
PROVIDENCE 177 Union St.

SOUTH CAROLINA
COLUMBIA Old Sears Bldg., 1001 Harden St.
GREENVILLE N. Hills S/C
N. CHARLESTON 5900 Rivers Ave.

SOUTH DAKOTA
SIOUX FALLS 1700 S. Minnesota at 25th

TENNESSEE
CHATTANOOGA 636 Northgate Mall
JOHNSON CITY Peerless Center
KNOXVILLE Cedar Bluff S/C, 9123 Executive Park Dr.
MEMPHIS 4665 American Way, 1997 Union Ave.
NASHVILLE 2115 Franklin Pike, Rivergate Plaza

TEXAS
AMARILLO Wellington Sq. S/C, 1619 S. Kentucky
ARLINGTON 2500 E. Randol Mill, Suite 113
AUSTIN 8764 E. Research Blvd., Southwood Mall, White Blvd.
BROWNSVILLE 1639 Price Rd. (Hwy. 77)
BEAUMONT 5330 Eastex Frwy.
COLLEGE STATION 2414 Texas Ave. South
CORPUS CHRISTI 1711 S. Staple St.
DALLAS 15340 Dallas Pkwy., Suite 1100, 2930 W. Northwest Hwy., 1517 Main St., 2588 Royal Ln.
EL PASO 9515 Gateway West, Kern Plaza Shopping Center, 3100 N. Mesa
FT. WORTH 231 One Tandy Center, 2801 Alta Mere
GALVESTON 5924 Broadway
HARLINGEN 1514 S. Hwy 77, Sunshine Strip
HOUSTON 211C-FM 1960, 10543 Gulf Fwy., 5900 North Fwy., 6813 SW Fwy., 809 Dallas St., Holland Square Center, 10920 East Freeway, (713) 453-0600, Champion Forest Plaza, Champion Forest Dr. and F.M. 1960 West, 1018 Gessner, 3278 South Loop West (So. Main at 610)
HUMBLE 19300 "B" Hwy. 59 (at FM 1960)
HURST Northeast Mall
IRVING 2011 West Airport Fwy.
LAREDO 102 East Calton Rd.
LUBBOCK 3625 34th St.
MESQUITE Town East Crossing Shop. Ctr.
ODESSA 1613 "A" East 8th Street
RICHARDSON Fleetwood Sq. S/C, 202 W. Campbell Rd. 320 S. Central Expr.
SAN ANTONIO 6018 West Ave., 4249 Centergate, Riverbend Parking Garage, 211 W. Market St. (Downtown)
WICHITA FALLS 1720-A 9th St.

UTAH
OGDEN K-Mart Shopping Center, 3672 Wall Ave.
OREM Grand Central Plaza, 384 East & 1300 South
MURRAY 6051 S. State Ave.
SALT LAKE CITY 301 South State St.

VIRGINIA
ALEXANDRIA 3425 King St. at Quaker Ln.
ARLINGTON Crystal City, 2301 So. Jefferson Davis Hwy.
FAIRFAX Westfair Center, 11027 Lee Hwy.
LYNCHBURG Hill's Plaza, Ward's Rd.
NEWPORT NEWS Newmarket South Shop. Ctr.
NORFOLK 5731 Poplar Hall Dr., Wards Corner, 122 E. Little Creek Rd.
RICHMOND Willow Lawn S/C, 1617 Willow Lawn Dr., 7728 Midlothian Turnpike
ROANOKE Franklin Bldg., 3561 Franklin Rd. S.W.
ROSSLYN 1911 N. Ft. Myer Dr. at Rt. 29

WASHINGTON
BELLEVUE Crossroads Mall, North East 8th & 156 St.
BELLINGHAM 1111 Cornwall Ave., Suite B & C
FEDERAL WAY 33505 Pacific Hwy. South
OLYMPIA 106 N. Wilson
SEATTLE 18405 Aurora Ave. N., 1521 3rd Ave., 50 Roosevelt Way NE
SPOKANE 7702 N. Division, E. 12412 Sprague
TACOMA 7030 S. Sprague
TUKWILA 15425 53rd Ave. S.
YAKIMA 1111 N. First St.

WEST VIRGINIA
DUNBAR, Dunbar Village Shop. Ctr.
HUNTINGTON 2701½ 5th Ave.

WISCONSIN
APPLETON 2310 West College Ave.
MADISON 57 West Towne Mall
MILWAUKEE 6450 N. 76th St., 729 N. Milwaukee (Downtown)
WEST ALLIS 2717 South 108th St.

Radio Shack

TRS-80 Microcomputer News
P.O. Box 2910
Fort Worth, Texas 76113-2910

BULK RATE
U.S. POSTAGE
PAID
Radio Shack
A Div. of Tandy Corp.

ADDRESS CHANGE
☐ **Remove from list** ☐ **Change as shown**
Please detach address label and mail to address shown above