THE R O M NEWSLETTER
USERS GROUP OF ORANGE COUNTY
17301 SANTA ISABEL STREET
FOUNTAIN VALLEY, CA 92708

8/89
Dallas TI Computer Group (DTIHCG)
PO Box 29863
Dallas,
TX 75229

```
ROMROMROMROMROMROMROMROMROMROMROMROMRO
O■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■M
M■OMROMROMROMROMROMROMROMROMROMROMRO■R
R■M■■■■■■■OMROMR■■■■■■OMROM■■MROM■■M■O
O■R■MROMR■■ROMR■■ROMR■■ROMR■■■OM■■■R■M
M■O■ROMROM■OMRO■ROMROM■OMRO■R■■■■M■O■R
m■M■OMROMR■MROM■OMROMR■MROM■OM■■MR■M■O
O■R■MROMR■■ROMR■MROMRO■ROMR■MROMRO■R■M
M■O■■■■■■■ROMRO■ROMROM■OMRO■ROMROM■O■R
R■M■OMR■■ROMROM■OMROMR■MROM■OMROMR■M■O
O■R■MROM■■MROMR■■ROMR■■ROMR■MROMRO■R■M
M■O■ROMRO■■OMROM■■■■■■ROMRO■ROMROM■O■R
R■MROMROMROMROMROMROMROMROMROMROMROM■O
O■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■M
MROMROMROMROMROMROMROMROMROMROMROMROMR
```

AUG 1989

SERVING THE TI 99/4A HOME COMPUTER COMMUNITY

WE MEET AT MERCURY

TIME AND PLACE OF MEETING

The FIRST Thursday of each month at

MERCURY SAVINGS and LOAN
7:30 PM

West of Beach at 7813 Edinger Ave., Huntington Beach, Cal. Use the WEST enterance. Park on the west side of the building. All are welcome.

```
 :     (SD) 405 FREEWAY        :
--------------------------------
 :                             :
G:         EDINGER AVE.        :B
O+-----------------------------+E
L:                             :A
D:  :   PARKING     +-------+  :C
E:  :    AREA       :MERCURY:  :H
N:  :               :SAVINGS:  :
 :  :               :       :  :
W:  :               +-------+  :
E:  :                       :  :
S:  :                       :  :
T:  +-----------------------+  :
 :                             :
 :                             :
 :                             :
 :                             +
 :                             :
--------------------------------
 :           HEIL AVE.         :
```

U.G.O.C. OFFICERS

PRESIDENT	JIM SWEDLOW	897-9209
VICE-PRES	JIM JOLLY	531-7782
SECRETARY	EARL RAGUSE	847-5875
TREASURER	JERRY RASH	631-0579
PAST-PRES	STAN CORBIN	892-2818

COMMITTEE CHAIRMEN

JIM MORRIS	MEMBERSHIP	546-8354
SILES BAZERMAN	NEWSLETTER	897-2868
BEN HATHEWAY	BULLETIN BOARD	751-4332

SOFTWARE LIBRARY

KNUTE ERSLAND....................842-0859

NEWSLETTER LIBRARY

EARL RAGUSE....(Phone for time)...847-5875

PUBLICATION NOTICE

The ROM is published by the Home Computer Users Group of Orange County, a non-profit organization. Permission is granted to similar groups to reproduce articles published in the ROM provided author and source credits are included. Opinions expressed in this publication are those of the author and do not necessarily represent those of the ROM, it's Editor, or the HCUGOC, it's officers or members.

We solicit letters and articles of interest to the TI-99/4A user community. Material accepted may be edited for fit and format. No payment is offered nor intended (other than your byline).

NEWSLETTER CONTRIBUTORS

NEWT ARMSTRONG	EDITOR EMERITUS
STU HAYNIE	PRODUCTION
EARL RAGUSE	TI FORTH
ADRIAN ROBINSON	ASSEMBLY
ROBERT SHAFFER	CIRCULATION
JIM SWEDLOW	AT LARGE
BILL NELSON	GRAPHICS

TI CLUB ACTIVITIES

CLUB	ACTION	DATE	INFO
BUG	GENERAL MEETING	SEP 04	532-1554
UGOG	GENERAL MEETING	SEP 07	897-9209
UGOC	LIBRARY, FTNVLY	SEP 11	842-0859
ET99	GENERAL MEETING	SEP 09	837-8757
UGOC	ASSEMBLY SIG	SEP 14	537-1839
UGOC	LIBRARY, FTNVLY	SEP 18	842-0859
UGOC	BOARD MEETING	SEP 21	897-9209
UGOC	ASSEMBLY SIG	SEP 28	537-1839
UGOC	NSLETTER LIBRARY	CALL	847-5875

FROM THE PRESIDENT
By Jim Swedlow

Well, the August meeting with TI Writer work shops went well. I think all (most?) of us enjoyed them and learned someting. Thanks to everyone who brought in systems and/or spent time with a group.

I can't tell you about September because we have not firmed it up yet but we are working on some interesting things.

A revision to our constitution and bylaws in in the works. Nothing special, we just want to change our operating rules to match what we do. Look for the changes in a month or so.

With any luck, there should be a major update to the disk library in the September meeting.

Future meetings under discussion are a member swap meet and a library special.

'Nuff said.

****
****

MEMBERSHIP CORNER
By Jim Morris

Our membership is currently at sixty seven. Welcome aboard to a new member Gordon Homstad, he joined our group in June. Five members are currently ninety days in arrears. Two say the check is in the mail. The other three are yet to be contacted. Eleven members memberships are currently due. Two members may be dropped if we don't hear from them by the end of the month.

We obviously have an attrition problem and somehow although we have a number of new prospective members very few actually join the club. Club ads placed in various computer publications are our main source of prospective new members.

Three new prospective members are expected at the August meeting. Some way has to be devised to increase our membership any bright ideas would be appreciated. To date I haven't received even one input regarding how we can increase our membership!

A large number of updated library lists have been passed out and as you know a large number of excellent programs are available so please patronize it if at all possible.

JULY BOARD MINUTES
By Earl Raguse

The meeting was held at the home of our Editor, Siles Bazerman. Those present were, Jim Swedlow, Jim Jolly, Earl Raguse, Siles Bazerman, Ben Hatheway, Stan Corbin, Bill Nelson, and Stuart Haynie. Jim Morris is still suffering from an infected foot.

There were no treasury nor membership reports.

The Secretary had only a recieved NL disk from the Tidewater UG to report.

The disk and newsletter librarians had little to report.

Siles was able to report that the ROM had already been printed by Stu and was given to Earl for labels and distribution.

Stan Corbin was elected to hold the office of Past President, in accordance with the By Laws. Our actual Past President is our current President.

The name of Jack Shugrue was added to our Hall Of Fame eligibles. Earl Raguse is to investigate if there is a suitible fund which has been set up for post-humously honoring Bill Kneckt for his great music.

Equipment will be provided for the TIW workshop by Jim Swedlow, Stan Corbin, Siles Bazerman, and Earl Raguse. Programs for Sept and Aug meetings were vaglely discussed.

We then got to the main thrust of the meeting, which was updating of the Constitution and By Laws. Many proposals were considered to make these documents more accurate and workable without changing the intent and/or actual practise.

All members will be provided with a copy of these proposals for review. They will be made a part of the ROM so that members can find out what is going on without actually attending a meeting. They must, however, attend a specified meeting if they wish to vote on them.

The meeting was adjorned at 9:2Ø, and Siles served some excellent refreshments. I say excellent, because he served coconut macaroons again!

The Dark Side

by Stan Corbin

Recently a preposterous theory about Dark has been espoused. Though the "theorists" claim the speed of Dark is 186,ØØØ miles per second, there is no data to substantiate this claim. In fact according to Ian D. Dark, professor of physics at B. S. college, no method of measuring the speed of Dark has been developed. Professor Dark states, "I have been working in the dark for years and am unable to substantiate the claims of Mr. Earl Raguse in his theory of Dark. In fact, the postulation that Dark is a combination of colors in phase opposition producing dc, is plain hogwash -unless dc means, doesn't compute". The great theoretical physicist Doan U. Bleevit, in agreeing with professor Dark, pointed out that the cue ball which is white, may sit in the shadow of the black eight ball and still remain white, while both are in the shadow of your fat frame. So much for Mr. Raguses reference to Dark remaining under the car or that your fat frame acts as a shield for Dark. In a bit of sarcastic humor, physicist Doan U. Bleevit suggested that Mr. Raguse has spent too much time behind the eight ball and so remains in the dark about the proven theory of light.

Mr. Raguses cohort in the "theory of Dark", Mr. Newt Armstrong, has the impertinence to suggest, that the reason the black lettering remains on a printed page when exposed to a "Dark sucker", is that the lettering is held in place by a special glue. This ridiculous suggestion is obviated by their own claim that Dark has a speed of 186,ØØØ miles per second. It is impossible to spread all that glue before Dark would be sucked up, regardless of how good one is at spreading it. Despite the obvious, Mr. Armstrong continues to stick to his glue theory.

Mr. I. C. Clearly, an optical engineer, when asked about the Dark theory, suggested that it would be very difficult for Mr. Raguse or Mr. Armstrong to explain Rainbows. He also raised the question of how they could possibly explain moonlight (moondark?), bearing in mind that the moon has been explored and found not capable of "sucking" Dark. After all, the moon remains a passive body and as the "theorists" have pointed out in their arguments, Dark travels in a straight line. That precludes Dark from being drawn from the Dark hemisphere of the earth when the Sun is extracting Dark from the opposite hemisphere. In other words, with the Sun diametrically opposite the Dark side of the earth, there is no way for Dark traveling in a straight line, as advocated, to suck Dark from the opposite side of the earth.

Then there is the phenomenon of sight, if Dark is being sucked up then how are we able to see? Dark would be sucked out of our eyes and since the Dark theory doesn't allow for light we would not be able to see. Of course this brings up an interesting subject what is it we see if Dark is sucked out of our eyes. How is it, when we close our eyes and look at the sun, we seem to have Dark sucked out of our eyes despite Mr. Raguses claims that Dark cannot be sucked through solids(our eyelids). To make his theory credible, will require a further epexegesis by Mr. Raguse.

When asked why he chose to publish in the public press (ROM) and not in the prestigious scientific journal Nurture, Mr. Raguse stated, "It was a theory which could not wait the annals of Nurture". It is believed however, that Mr. Raguse was pre-empting some other "scientist" from shedding Dark on light or light on Dark or would that be Dark on Dark ".

Mr. Raguse denies the claim, that he is attempting to resurrect the Dark Ages, as totally false and that that claim was instigated by some lightheaded, blackhearted derogator.

Just who is this Earl Raguse? Some claim he is the Prince of Dark, while others say he is Dark Fader, the representative of the Dark Forth. Whoever, or whatever, he has given us some Dark thoughts.

A sobering thought comes to mind, what if Mr. Raguse is correct. Can you imagine all the changes in language and customs that would occur? Can you see yourself, asking for a Millers Dark or Buying a Dark bulb for your lamp? If you lost weight you'd be getting Darker rather than lighter, and what of the Goodyear blimp being referred to as a Darker than air craft.
The day Mr. Raguse brought his Dark theory to light, was a very Dark day indeed.

****
****

EDITORS COMMENT

I think we have kept you in the DARK long enough. This will, for the present time, conclude our excursion into "scientific" areas besides Computer Science.

TI BITS * Number 29

By Jim Swedlow

I had a problem recently making TI Writer print out some revisions to our club bylaws and I though you might be interested. Some background is needed first.

TALKING TO YOUR PRINTER

Some time ago I mentioned that there are almost no standards for software codes to control printer functions (font, appearance, line spacing, etc). There are, however, a few things that are uniform. This month's effort centers on two of them: carriage return and line feed.

For simplicity, I will refer to them as <CR> and <LF>. <CR> is ASCII code 13 and <LF> is 1Ø (or, in hex, ØD and ØA, respectively).

What do they do? Just what they say. A <CR> causes your print head to return to the beginning of the current line. A <LF> causes the print head to advance one line, in the same print position. Sending both <CR><LF> causes the print head to move to the beginning of the next line.

This is almost universal. Many printers, however, have the ability to add a missing <CR> or <LF>. When in this mode, receipt of either command causes the print head to move to the beginning of the next line. Receiving both causes a double space. Usually this is controlled with DIP switches and is discussed in the printer's manual.

When you open a printer as "PIO" (I won't discuss RS232 names here because they act just the same), your RS232 card sends your printer a <CR> and a <LF> after every print line. You cannot control this as it is automatic.

You have some software control, however, by adding ".LF" or ".CR" to the printer name.

Here is a simple program that will show you what these extensions do:

```
1ØØ DATA "PIO", "PIO.LF",
    "PIO.CR"
11Ø FOR I=1 TO 3
12Ø READ A$ :: OPEN #1:A$
13Ø PRINT #1:A$:A$
14Ø CLOSE #1
15Ø OPEN #1:"PIO"
16Ø PRINT #1: : :
18Ø CLOSE #1 :: NEXT I
```

When the printer name is PIO, your RS232 card will add a <CR> and <LF> to each line and your printer will print:

```
PIO
PIO
```

When the printer name is PIO.LF, you RS232 card will only add a <CR> so you will get:

```
PIO.LF
```

but it will be printed twice (bold). With PIO.CR, neither <CR> nor <LF> is sent so you should have:

```
PIO.CRPIO.CR
```

PRINTERS AND TI WRITER

Wait, you say, in TI Writer (or FUNNELWEB, etc), I can add <CR> to lines. Does this change things? The answer is yes and no.

The primary function of this <CR> is to tell TI Writer that you have reached the end of a paragraph. It uses it in the Editor when you cause a reformat and in the Formatter when you use FIll or FIll and ADjust.

What does it do to your printer? Well, lets say you have this text in your Editor:

```
UGOC<CR>
```

and then print it with PrintFile and PIO as your printer name. Your TI Writer would send this to your printer:

```
UGOC<CR><CR><LF>
```

In other words, it adds a <CR><LF> to whatever is on the line. The double <CR> is the same as a single <CR> as the print head can only move to the beginning of the print line once.

WHY DOES THE FORMATTER USE A ".LF"?

In short, the answer is to enable you to use bold (@) and underscore (&). Remember, when your printer name is PIO.LF, your RS232 card only adds a <CR> to each print line. Here is how it works.

If your entire line was:

```
@UGOC
```

The formatter would send the following to your printer:

```
UGOC<CR>
UGOC<CR>
UGOC<CR><LF>
```

This would cause triple printing of UGOC or bold. Underline is similar. If your entire line was:

```
&UGOC
```

The formatter would send the following to your printer:

```
UGOC<CR>
____<CR><LF>
```

On the page, UGOC would be underlined.

STRIKE OUT AND ITALICS

I am almost done with the background. As I said at the beginning, I was working on an update to our bylaws. The normal way to show changes is to strike out old text and put new text in italics. For italics, I used TransLiterate to change the open bracket "{" to start italics and the close bracket "}" to stop italics:

```
.TL 123:27,52
.TL 125:27,53
```

Whenever the Formatter runs into an open bracket "{" or ASC 123, it will send the ASC codes 27 and 52 to the printer, which switches to italics. Similarly, "}" or 125 causes the Formatter to send 27 and 53 which returns my printer to its normal font. It made editing easy as anything I wanted in italics I simply enclosed in brackets like {print this in italics}.

For strike out, I used transliterate to change the underscore to a dash:

```
.TL 95:45
```

Then, when I used the underscore command, the Formatter printed strike out. So &old &text would print as:

~~old text~~

NOW FOR THE PROBLEM (FINALLY)

When I mixed strike out and italics on the same line and the italics were first, I got garbage. Why? Because TI Writer and my printer count characters differently.

Suppose I had this line:

{new text} for &old &text

When it went through the formatter, the printout looked like this:

new text for old text----

Why? Well, you see, TI Writer counted the open and close brackets as characters while the printer, recognizing them as control codes, didn't. So TI Writer thought it was sending this:

```
..new text.. for old text<CR>
             --- ----<CR><LF>
```

The double dots ".." stand for the control codes that the brackets were translitterated into. TI Writer, as you can see, counts the number of characters sent to the printer, not the number in the Editor print line.

The problem was that the printer counts the print characters this way:

```
new text for old text
             --- ----
```

The result was that the strike out was not correctly aligned.

The solution was to make sure that, in the same line, italics did not precede strike out. It could follow or be on a different line, but not precede.

So I wrote this:

&Old &Text becomes {new text}

and it printed like this:

~~Old Text~~ becomes *new text*

I left ADjust off (using only FIll) because this difference in character counts defeats the way the Formatter right justifies text.

Enjoy.

****

(WRITE RIGHT continued from Page 9)

Again press F9, P(rint)F(iles), <ENTER>, and this time for filename put in PIO (or whatever your printer command is). You will now have a hard unformatted copy of LEARN. You can go through this sequence one more time, but this time use the filename LPIO. You should have a copy including line numbers.

I see that I'm running out of time and space so I'll close for now. To exit the program press Q(uit), <ENTER>. You might have noticed that the Q command gave you a choice of Save, Purge, or Exit. Purge will clear the buffers and leave you in command mode, while Exit will terminate the program and take you back to the menu screen. Save is another chance to save your program, but since we already did that, just press E <ENTER>.

Next time we will edit LEARN and see how simple it is to make changes in text. Until then, WRITE RIGHT.

CALENDAR PROGRAMS

by Adrian Robinson

AND NO ASSEMBLY

This article has been in the back of my mind for several years. Calendars are a fairly frequent subject of programmers' efforts and, when new calendar programs appear, I often feel an urge to comment on them. Occasionally, a program may be founded on an incorrect algorithm, which results in inaccuracy or in a limited range of application. More often though the algorithm is correct, but extremely inefficient programming techniques are employed, which result in a program much longer than it needs to be. A calendar program can really be quite simple and I will discuss some simple techniques, but first I would like to say a little about the foundations for, and history of, the calendar. The history of development of the calendar is an interesting subject in itself.

A calendar is basically just an enumeration of the days in a year, or of the number of rotations of the Earth about its axis during one revolution of the Earth about the Sun. Oddly enough there are at least three different definitions of the length of a year. They are the Sidereal Year, the Anomalistic Year and the Tropical Year. We will not discuss the first two of these but the Tropical Year is the year of the seasons and thus the basis of the calendar. The Tropical Year is the time between two successive occurrences of the Vernal Equinox, the time when the Sun's apparent motion crosses the Earth's equatorial plane in its northward movement, and marks the beginning of Spring. The length of the Tropical Year is approximately 365.2422 Mean Solar Days. We will refer to this value later.

As civilizations developed in various parts of the world, some of the earliest recorded information includes astronomical observations and the development of a calendar in one form or another. An observer may notice that the position of sunrise or sunset on the horizon varies, moving north and south, during the year and he may relate that cycle to the seasons. Then, by counting the days in the cycle he can forecast the seasons. A hunter-gatherer society can now decide its migration cycle or an agricultural society can decide when to plant and when to harvest its crops. Thus, even a crude calendar can be a powerful tool.

The roots of our calendar go back more than 7ØØØ years. The ancient Egyptians first determined the length of the year as 36Ø days and then, centuries later, as 365 days. With this value, in the year 4236 b.c., they devised a calendar of twelve thirty-day months supplemented by five consecutive year-end holidays. Not a bad idea!! In fact, I think that I would prefer this to our current calendar with its months of oddly varying lengths. Much later, they realized that the year was more nearly 365 1/4 days in length and, in 238 b.c., they provided for a leap year every fourth year. This calendar, however, was not widely accepted.

Meanwhile, the Romans were struggling still with a lunar calendar of twelve alternating 29 and 3Ø day months.(The cycle of lunar phases is 29.5 days). But twelve lunar months total only 354 days so they found it necessary to throw in an extra month about every two or three years to keep the seasons more or less in order.

Finally, in 46 b.c., Julius Caesar set about reforming the Roman calendar. He made the calendar independent of the phases of the moon and changed the lengths of the months to total 365 days. He also adopted the Egyptians' 365 1/4 day year, adding a day to February every fourth year. In addition, he restored the time of the vernal equinox to its ancient date of March 25. Naturally, this became the Julian Calendar. Aside from some inconsequential "political" adjustments, the only significant change in the next sixteen centuries was the introduction, early in the fourth century a.d., of the Christian seven-day week. This added a further complication to the calendar since the year is not divisible by seven.

Now the Julian calendar of 365.25 days differs from the actual length of the Tropical Year of 365.2422 days by about one day every 128 years. As a result, by the sixteenth century, the calendar date of the vernal equinox had advanced from March 25 to March 11. If this were to continue long enough, we would have Spring in December and Winter in August.

In 1582, therefore, Pope Gregory XIII proceeded to correct the errors of the Julian calendar. You may wonder why Pope Gregory would be concerned with the calendar. Well, the date of Easter was prescribed in a.d. 325 to be the first Sunday on or following the day of the first full moon following the vernal equinox. Hence he wanted to maintain a constant date of the vernal equinox. He therefore ordained that century years would not be counted as leap years unless they were also divisible by four hundred. This has the effect of making the average length of the calendar year

$$365+1/4-1/1ØØ+1/4ØØ = 365.2425 \text{ days}$$

which is a much better approximation to the true length of a year. He also made March 21 the date of the vernal equinox, as it was in a.d. 325 instead of the original March 25. Thus, in 1582, in order to correct the calendar, October 5 suddenly became October 15.

I think that it is worth noting that two contemporaries of Pope Gregory were Copernicus and Galileo, both of whom were persecuted by the church for their "heretical beliefs" that the earth revolved about the sun and was not the center of the universe.

The world's acceptance of the Gregorian calendar has been slow. It took Great Britain and her empire two centuries to accept the Gregorian calendar(1752). It seems that the venerable old Church of England has been very influential and it was not anxious to accept a calendar that was ordained by the Pope of Rome. However, it is generally accepted today, at least in the western world. Although the Gregorian calendar has a residual error of one day in about 33ØØ years, it is a good enough approximation for many more centuries of practical use.

From time to time proposals are made for reform of the internal structure of the calendar. Although it still has twelve months, there is no longer any direct connection to the phases of the moon and the varying lengths of the months are actually illogical. The seven-day week, however, has become an integral part of everyday life and would be very difficult to change.

My personal preference would be a year of thirteen identical months of twenty eight days, four weeks, each. Every month would begin on Sunday and end on Saturday. The year would be completed with an extra holiday (New Year's Eve?) following the thirteenth month. Leap years would have an additional extra holiday at the end. The current year, 1989, would have been a good year to start this new calendar since it already started on a Sunday. The vernal equinox by the way, would occur on the 24th day of the third month, whatever its name might be. I am sure that there would be very strong resistance from the calendar publishing lobby since this would be a perpetual calendar. The calendar would be identical for every year. There would never be a need to buy a new calendar except to get a new set of pictures.

In addition to the above reform, I would add a new term to the formula for the mean length of the calendar year:

```
365+1/4-1/1ØØ+1/4ØØ-1/32ØØ
=365.2422 days(rounded)
```

The new term, -1/32ØØ, means that years divisible by 32ØØ would not be counted as leap years, whereas in the Gregorian calendar they would be. The agreement of this value with the true length of the Tropical Year would allow this Robinsonian Calendar to serve without error for about 1ØØØ centuries.

But wait!! This article was supposed to be primarily about calendar programming techniques. I guess I just got carried away. Let us reduce the calendar to its fundamentals. As indicated above, the calendar year is basically just a string of numbers from 1 to 365 (or 366). The division into months simply converts it to a concatenation of strings from 1 to 31, 1 to 28, etc. with allowance made for leap years. Introduction of week days results in a need to "register" the date string to the repeated weekday string. But this simply requires that we determine the "offset" of Jan. 1 from the first Sunday of the year. The rest of the year takes care of itself. If we think about these statements for just a moment we should see how very simple the problem is. The names and lengths of the months, as well as the weekdays, are predetermined. All we need to define a calendar, for any year, are two items of data:

1. What is the offset for Jan. 1?
2. Is it or is it not a leap year?

The included program, for convenience, actually computes the offset for each month individually. This way, a slight modification will allow the program to compute and print a single month.

So, finally, let us take a look at the program. Lines 18 and 2Ø contain the names and lengths of the months. Line 22 forms a string of weekdays and an underline. Line 24 forms a string of dates 1 to 31 in four-character blocks. The program manages to take advantage of Basic's twenty-eight column screen. Note the printer OPEN statement in line 28.

The mainline program starts with the loop at line 3Ø, which reads the twelve months' names and lengths and calls the subprogram MONTH. MONTH then computes the number of days elapsed from a hypothetical zero date to the first day of the current month, allowing for all leap years. Line 48 then converts that to a modulo 7 number(Ø-6) which is the number of weekdays preceding the first date of the month. Line 32 adds a day to February if Y is a leap year. Line 34 uses L to take the proper segment of M$, and B to add the proper number of blanks to the front of M$ for the offset. Line 36 then displays and prints the month.

Lines 3Ø through 5Ø, just twelve lines, comprise the essence of the program and it computes, displays and prints an accurate calendar for any year later than 1582, with six months per page. With relatively minor modifications, the program could be tailored to print in any desired format.

```
10 ! CALENDAR PROGRAM
12 ! by Adrian Robinson
14 !
16 ON WARNING NEXT :: CALL C
LEAR
18 DATA JANUARY,31,FEBRUARY,
28,MARCH,31,APRIL,3Ø,MAY,31,
JUNE,3Ø
2Ø DATA JULY,31,AUGUST,31,SE
PTEMBER,3Ø,OCTOBER,31,NOVEMB
ER,3Ø,DECEMBER,31
22 W$=" SU  MO  TU  WE  TH
FR  SA "&RPT$("-",28)
24 M$="" :: FOR I=1 TO 31 ::
 M$=M$&RPT$(" ",3-LEN(STR$(I
)))&STR$(I)&" " :: NEXT I
26 INPUT " YEAR: ":Y :: IF Y
<1583 THEN PRINT :"YEAR MUST
 BE LATER THAN 1582": : :: G
OTO 26
28 OPEN #1:"PIO",DISPLAY ,VA
RIABLE 28 :: PRINT #1:Y: :
3Ø FOR M=1 TO 12 :: READ MO$
,L :: CALL MONTH((Y),M,B)
32 IF M=2 THEN IF (Y/4=INT(Y
/4))-(Y/1ØØ=INT(Y/1ØØ))+(Y/4
ØØ=INT(Y/4ØØ))THEN L=L+1
34 D$=RPT$("    ",B)&SEG$(M$
,1,4*L)
36 PRINT " ";MO$:W$:D$: : ::
 PRINT #1:" ";MO$:W$:D$: :
37 IF M=6 THEN PRINT #1:CHR$
(12): : :
38 NEXT M :: CLOSE #1 :: END
4Ø SUB MONTH(Y,M,B)
42 F=365*Y+31*(M-1)+1
44 IF M>2 THEN F=F-INT(2.3+.
4*M)ELSE Y=Y-1
46 F=F+INT(Y/4)-INT(Y/1ØØ)+I
NT(Y/4ØØ)
48 F=F-1 :: B=F-7*INT(F/7)
5Ø SUBEND
```

****
****

(MULTIPLAN continued)

e. The command menu ...
   1) Press "G" for "Go to".
   2) Press "R" for "Row-column".
   3) Type the row number to go to.
   4) Press "Ctrl" + "A" to go to next field.
   5) Type the number of the column to go to.
   6) Press "enter" to process the command.

8. SETTING COLUMN WIDTH:

a. Move the cursor to the column which will be changed.
b. Press "F" for "Format".
c. Press "W" for "Width".
d. Enter the number for the new width of this column.
e. Press "enter" to process the command.

9. FORMAT A CELL (displays Alpha info in adjacent cells)

a. Move cursor highlight over the cell.
b. Press "F" for "Format".
c. Press "C" for "Cell".
d. Press ":->->->" (arrow over & down).
e. Press "CTRL" + "A" (TWICE).
f. Press "C" for "Continuous".
g. Press "enter" to process the command.

1Ø. EDITING A CELL:

a. Move cursor highlight over the cell.
b. Press "E" for "Edit".
c. Press "Function" + "4" to move the cursor right on the displayed text without deleting characters.
d. Press "Ctrl" + "4" to move the cursor left on the displayed text without deleting characters.
e. Press "Function" + "9" to delete characters.

11. UPDATING AND SAVING:

a. When you change any number in one of the cells included in formulas, all dependent cells are automatically updated to reflect the change.
b. After making desired changes to the spreadsheet, it is important to SAVE the spreadsheet to the data diskette.
   1) Press "T" for "Transfer".
   2) Press "S" for "Save".
   3) Press "enter" to process command.

12. WINDOWING

a. Place cursor in the first cell with data. (To the RIGHT of the verticle title cells & BELOW the horizontal title cells.
b. Press "W" for "Windows".
c. Press "S" for "Split".
d. Press "T" for "Titles".
e. Press "enter" to process the command.
f. To cancel the WINDOWING:
   1) Press "W" for "Window".
   2) Press "C" for "Close".

- ENJOY, HAVE FUN -

Call me Monday thru Friday 9 to 11 am
for any serious questions at:

(714) 898-2Ø94

WRITE RIGHT #2

by Siles Bazerman

Something I forgot to mention last time was the need for a control or function strip in the slot on top of your console. If you purchased TI-WRITER new you should have gotten one with your package, but if you have TK-WRITER or are missing the strip I will list the keys so you can make one of your own.

	CONTROL	FUNCTION
1	OOPS!	DEL CHAR
2	REFORMAT	INS CHAR
3	SCREEN COLOR	DEL LINE
4	NEXT PARAGRAPH	ROLL DOWN
5	DUPE LINE	NEXT WINDOW
6	LAST PARAGRAPH	ROLL UP
7	WORD TAB	TAB
8	NEW PARAGRAPH	INS LINE
9	NEW PAGE	COMMAND/ESC
Ø	WORD WRAP	LINE NOS.
+	NOT USED	QUIT

I will talk about each of these as we need to use them, and also give you the alternate keys, if available.

Well, thats enough talking about the program. Let's actually use it now. Before we start, though, you will need to have an initialized disk on hand to store your work. NEVER use your program disk to store files on. In case of any trouble or mistake you may wipe out all the program material. To start, for TI-WRITER insert your command module and turn on the computer, while for the rest just turn on the computer. Now insert the program disk (in DSK.1 if you have more than one drive) and your data disk in any other available drive. If you have only one drive then keep the data disk handy. TI-WRITER etc are self-booting. For TI-WRITER press the key for TI-WRITER and for others select EXTENDED BASIC. After loading you will see a title screen with: 1.EDITOR; 2.FORMATTER; 3.UTILITY, or a menu. The exact words will be different between all the programs, but in any case we want to select 1.EDITOR. This is the program where you do the creating, editing, and any formatting or other things to your text.

The Formatter program is where the machine takes over and does its job of producing a finished document. The Utility choice is for other supporting programs; I use mine for an auto spell-check by Dragonslayer.

Press key 1 and wait for the Editor to load. When ready you will see on the screen a line that reads Edit, Tabs, Files, Lines, SearcH, RecoverEdit. You are now in the command mode and will note that capital letter(s) are the commands to enter. Before going any further let's use the CONTROL 3 key (screen color). Press this combination and note the screen colors, (both foreground and background) change color. Select one you like, or more important one that your eyes like.

Now press T(abs), then <ENTER>. You will see two lines that read: TABS and a series of numbers, and a second line that has L,and T's. Press FUNCTION 5 (Next Window), the screen will change and another press of F5 will show the last window. Another will return you to the original screen. This command (T) will let you set your margins , tabs, and if you want your paragraph indentations for both the screen and any printing in the EDITOR mode. The next window command allows you to see 4Ø columns at a time: first, Ø to 39; second, 2Ø to 59; and third, 4Ø to 79 (8Ø columns).

The default setting for Tabs is left margin Ø, right margin 79, and tabs at 5,1Ø,15,25,35,45,55,and 65. These are fine if you don't mind flipping back and forth through windows to read your text, but this drives me crazy. I use a setting of Ø, L(eft margin), and 39, R(ight margin), for margins, tabs at 1Ø and 15, and I(ndent paragraph) at 5. To enter these just use the space bar to erase what you don't want and type in the appropriate letter where you want it. I also eliminate the line numbers by using Function Ø (Line nos.). This gives me a full screen of text easily readable without going back and forth through windows. Don't WORRY about the finished document margins. These will be set when we print using the Text FORMATTER. For learning purposes let us all use L at Ø, R at 32, and I at 5 so everything will correspond.

We are now ready to write so press E(dit) and <ENTER>. You are in the edit mode for writing, changing, or correcting your work. Copy LEARN from the last ROM exactly as written,spelling errors, bad spacing, repeat lines, and all. We will be correcting these as we go through all the EDIT commands.

If you are all through with the copying, we can go on. First, lets save this material so we don't have to do it over. Press F9 (Command/Escape) to return to the Command mode, then press S(ave)F(iles), <ENTER>. The screen will ask for a file name. Type in DSKx.LEARN, and <ENTER>. When this is saved you will find that you are back in the EDIT mode.

(continued on Page 5)

MULTIPLAN DEMONSTRATION
By: Kathryn D Coonley and Erwin H Metz

1. PRELIMINARY START UP:

NOTE: TI Memory Expansion and Peripheral Expansion System is needed with card type peripherals (i.e. fully loaded P-Box).

a. Insert Microsoft Multiplan TI Solid State Cartridge into the TI99/4A computer console.
b. Turn on the peripheral equipment (i.e. Disk Drives, Printer, etc.). Turn on the Monitor and Console.
c. Press any key on console to make the master selection list appear and press "2" to select Multiplan from the menu.
d. Select the screen color schemes (12 selections) by depressing the "space bar" on the console until your selection occurs.
e. Insert the Multiplan diskette into Disk Drive #1.
f. Press "enter key" on the console and the Multiplan Overlay will appear on the monitor screen.

2. DATA TYPES:

a. Cells - On a spreadsheet information is entered into "cells". A typical cell is 1Ø to 3Ø characters long, so several cells fit on one printed line. A whole column of cells have the same width, and that width can be changed.
b. Once a number has been typed into a cell, it can be added, subtracted, multiplied, etc. to other numbers in other cells. If the number is too big for the width of cell, the number will not show (rather a # symbol or an exponential value will be displayed) until the column width is increased (see step #8 below).
c. Alpha characters act differently. If more characters are typed than the width of the cell, the number of characters equal to the width of the cell are displayed. All the characters can be displayed in the succeeding cells of the row by using special format commands (see step #9 below).

3. SPREADSHEET COMMAND MENU:

a. The menu at the bottom of the screen, (including submenus obtained after the initial menu selection) permits access to all Multiplan features, to include:
    1) set up the initial spreadsheet
    2) change entries on the existing spreadsheet.
    3) print out the spreadsheet (total or partial)
    4) save the spreadsheet to the data diskette in a selected drive
    5) delete an existing spreadsheet
b. The commands can be selected by typing the first letter of the command, or pressing the space bar to move the highlight over the command and pressing enter.
c. Press "Ctrl" + "C" (at the same time) to return to the main menu from any submenu.

4. SET UP FOR 2ND DISK DRIVE (Data Disk):

a. Select "Transfer" by pressing "T".
b. Select "O" for "Options" from the submenu that appears.
c. Press "Ctrl" + "A" (at the same time) to move to the next field.
d. Type "DSK2" next to "SETUP:"
e. Press "enter key" to process the completed command.

5. LOAD FILE:

a. Insert data disk into drive 2.
    NOTE: Systems with only one disk drives should switch the data diskette with the Multiplan diskette located in drive 1.
b. Press "T" for "Transfer".
c. Press "L" to select "Load" from the submenu that appears.
d. To show all the files available to load on disk 2: press an "arrow key" ("Function" + "E,S,D,or X") OR press "Ctrl" + "1".
e. Use the "arrow keys" to move the highlight down to the file you want to load.
f. Press "enter" to load the highlighted file.

6. TO SPEED PROCESSING:

a. Press "O" for "Options".
b. Press space bar to move lighlight over "No".
c. Press enter to process the command.
d. Formulas will no longer recalculate after every change (a very slow, frustrating mode to be in). Instead they automatically calculate just before saving. Also you can manually start a recalculation by pressing "FCTN" + 8". Do this before printing to be sure you have the latest numbers.

7. MOVING AROUND THE SPREADSHEET:

a. Special keys ...
    1) "Ctrl" + "1" goes to upper left corner.
    2) "Function" + "1" goes to the lower right corner.
b. Use arrow keys ...
    1) "Function" + "E,S,D or X" will move the highlight around the spreadsheet.
    2) "Control"+ "E,S,D or X" moves up, down, or over a screen at a time.

(continued on page 8)