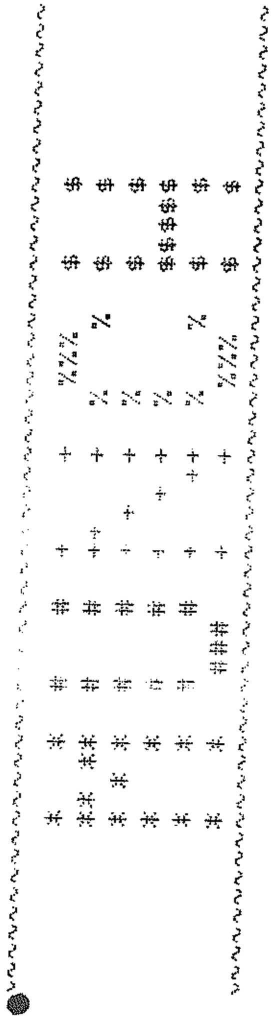
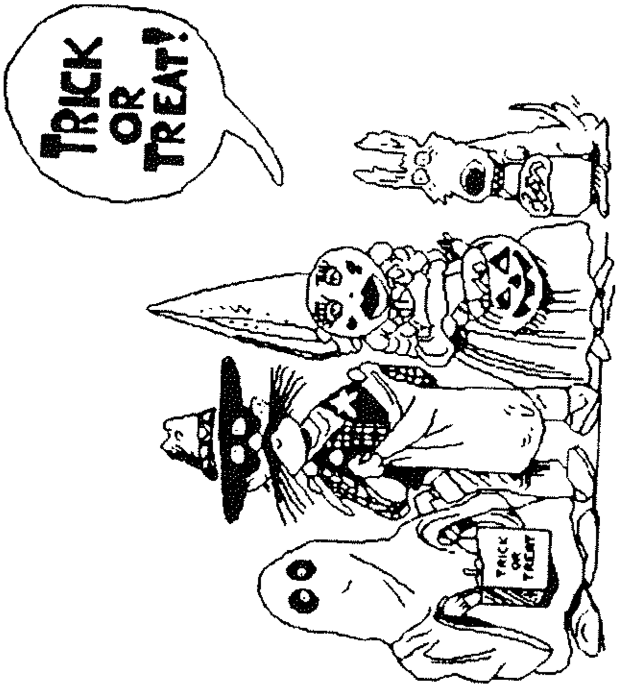


Mass Users of the Ninety-nine and Computer Hobbyists

OCTOBER 1994 Monthly Newsletter Version 13.10



M.U.N.C.H.
C/O J. W. COX
905 EGDEBROOK DRIVE
BOYLSTON, MASS. 01505

NEXT MEETING: TUESDAY, OCTOBER 11th.

POSTMASTER: Forwarding and Address Correction Requested.

FIRST CLASS!!

R.A. BISHOP
16 FRENCH AVE
NORTHCOTE 3070.
VICTORIA AUSTRALIA

NEXT MEETING TUESDAY, Oct. 11, 1994 7:00 PM. Happy Halloween!!!!!!!

MUNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

PRESIDENT	W. C. Wyman	865-1213
VICE-PRESIDENT	Bruce Willard	852-3250
TREAS./EDITOR/CLK.	Jim Cox	869-2704
DEMO LEADER	Jack Sughrue	476-7630
Asst. Demo Leader	Lou Holmes	617-965-3584
LIBRARIAN	Walt Nowak	413-436-7675
Advanced Programmer	Dan Rogers	248-5502

MUNCH DUES:

New Membership	$25.00
Renewal	$15.00
Newsletter Sub.	$13.00

SEPTEMBER MEETING. The September meeting had eight members in attendance. We tried out the disk of the month and the racing game was the big hit. Corson brought his Mac to the meeting and demoed a CD-rom player and interactive game. It was very interesting as Walt and I were talking about CD-roms before the meeting. Art won the raffle. Watch for the October issue of Micropendium. Charles Good does a review of Adventure II and the Tony Falco Classics, he has very nice things to say about both of them and his kind words are appreciated.

OCTOBER MEETING. If Jack can attend he will continue with the Funnelweb demo along with other interesting stuff. We will also try out the disk of the month.

RAFFLE. Every month we have a raffle to help defer the rental cost of our meeting hall. A typical raffle will have programs, blank disks, books, bumper stickers and all sorts of odds and ends of interest to the T.I. user. This month we have some Tandy Model 4 computers.

REPRINTS. Reprints are permitted as long as credit is given to M.U.N.C.H.

ARTICLES. I am always looking for articles for this newsletter, anything which interest you will probably interest other members of the T.I. community, so please share your ideas and opinions with all of us.

DISK LIBRARY. The disk library is at all meetings. We have copies of all disks in the library and they are available to members for just $1.00 for each disk unless otherwise specified. You can order them through the mail, please add $1.00 for the first disk and $.40 for each additional disk ordered to cover postage and handling.

DISK OF THE MONTH. This month's DOM #137 is the GPL #19 which contains E.T., Moonmine, Munchmobile and Sneggit.

ADVENTURE II. This is our fund-raiser for 1994/95. The cost to members is $4.00 add $2.00 for first class postage. The regular price is $6.95 plus postage. This is a two DSSD disk set, archived. There is also a special on The Adventure Compendium and Adventure II for members it is $8.00 plus $3.00 for first class postage.

PLAYING WITH NUMBERS 8 (cont)

infinite periodic fractions of their reciprocals which have no place in a numerical universe of whole numbers.

We can find it only in the reciprocal universe where the identity of prime numbers, like nucleons, atoms and molecules, is revealed in the wave-lengths and frequencies of their propagation. This is, to me, an exciting new frontier for exploration. It could never be done without computers. This is why it was never done in past centuries.

Now, let us take a look at some of the features of the program.
The opening menu offers the following choices:

(1): a NEW NUMBER with options of factoring decimal numerals no longer than 14 digits, or hectokilorad numerals of 5 decimal digits each. (The latter are the PBA "Power Blocks" that allow the factoring of large numbers). The limits are those of your computer, peripherals, DIM statements, or personal talents for modifying programs.

(2): a REP NUMBER (REPeatedUNITS), strings of ones automatically generated and properly converted into hectokilorad blocks, each a single base-10^5 numeral containing five decimal digits, (a relief from typing in numerals). An added bonus: since all numbers are contained in repunit strings as factors, the name of a side-game is to extract and manipulate the decimal numbers without being obliged to type them.

(3): a call for TEXT which permits you, when recording to printer as well as screen, to intersperse notes, comments or stray thoughts for later rediscovery.

(4): an EXIT when you want it. And:

(5): The oddballs, the mates of the Rep Numbers: (10^Y+1), tubes of Y-2 zeroes sealed off with a "1" at each end. The largest of these that you can enter for factoring under decimal auspices is 10000000000001.

I find it amusing to generate larger ones by entering 1,0,0,0,0,...1,END as a NEW NUMBER for hectokilorad "Power Block" factoring. It is unnecessary to enter leading zeros. 1,0,0,1 will come out on screen and paper as 1'00000'00000'00001. But these magnitude indicators will be generated for you if you enter "25" (not 5) of the menu. You will be asked to enter A,B where A is the power (10^A) of the first "1" and B, the last digit.
For rep-mates this is "1". For other series it will be other numerals.

$R(Y)=(10^Y-1)/9$, strings of Y "1"s. When this is multiplied by (10^Y+1) the result is $R(2Y)$. So all the factors of both $R(Y)$ and (10^Y+1) are also factors of $R(2Y)$ and all its multiples without end. The factors of (10^Y+1) are also factors of $(10^{3Y}+1)$ and its multiples.

With these, and other clues, composite numbers and some of their factors can be predicted, way beyond our ability to write them out. By work-playing with smaller numbers within the range of our program and the TI99/4A we may fashion a prognostic net for testing deeper waters for larger fish.

Next month's article will contain experimental examples for you to try out.

```
1112345678900987654321111
1                       1
1 PLAYING WITH NUMBERS  1
0         No 8          0
9  By Meredith Beyers   9
9                       9
9998765432100123456789999
```

THE "NUMRES" PROGRAM

The program file NR92/86/V3 is the latest version of a program that had its origin in programs written by me for my TI programmable calculators before the TI-99/4A computor was born. The purpose of the program was, and still is, to implement my own needs, and hopefully those of others, in exploring the numerical universe as a hobby.

Specifically this program contains the minimum of utility operations necessary to experiment with "Number Research" as summarized in the introduction to "Reciprocal Repetend Analysis". (The Computer Voice, Jan. 1992)

Nobody can be expected to understand all this, or engender any enthusiasm for exploring a numerical universe that is certainly "infinite" and apparently in utter chaos. A simple and exciting overview is urgently needed.

The foundation and infrastructure of a numerical universe consists of so-called "Counting Numbers", whole numbers, positive numbers. There are no negative numbers or fractions in it. These elements of chaos are the complexities of man's ignorance.

The basic structured numerical universe grows from a seed. "In the beginning was the Mark, and the Mark was one (1)." The counter, "1", is the seed that creates a universe by uniform incrementation: 1+1+1...... Names and symbols and notation systems have been invented to identify one number from another. There is no chaos in its creation. Every number is greater by one in the order of growth than the number preceding it, and one less than the one ahead.

Early explorers discovered that these numbers are alternately "odd" and "even", that odd+even=odd, but odd+odd and even+even are both even. They discovered that certain numbers generated by counting "things" could not be divided equally among a group, no matter how many were in the group or how many things there were to pass around. Coins, fruits, jewels, if possessed in these disturbing quantities could not be distributed in equal piles except by placing each one by itself. Generations later these odd disturbing numbers were called "Prime".

Primes did not complicate a numerical universe. They raised havoc in the minds of men.

Every other number in the universe could be broken down into a unique set of the pesky primes, but the primes, themselves unique with no duplicate, offered no identification but themselves. They defied the minds of men to "compute" them. Their identity is their function and their function is hidden in the

```
+PC))+5)&"DSK1."&A$(I-64+PC)
450 ! LOAD PROGRAMS
460 FOR I=0 TO 100
470 J=J+1 :: INPUT #1:A$(I),
B,C,D :: IF I=0 THEN 500 ELS
E IF J>=127 OR LEN(A$(I))=0
THEN 550
480 IF A$(I)="LOAD" THEN 470
490 IF A$(I)="LOADER100" THE
N 470
500 IF I=0 THEN DISPLAY AT(1
6,16):A$(0):: GOTO 530
510 IF (ABS(B)=5)OR(ABS(B)=4
 AND D=254)THEN DISPLAY AT(2
0,10):USING "### ##########
Y":I,A$(I)ELSE 470
520 IF Q$="N" THEN ACCEPT AT
(20,25)BEEP SIZE(-1)VALIDATE
("YN"):QQ$ :: IF QQ$="N" THE
N 470
530 NEXT I
540 ! :"X-EX BASIC"
550 EN$(0)=CHR$(181)&CHR$(19
9)&CHR$(10)&"X-EX BASIC"&CHR
$(0)
560 ! :"X-EX BASIC " "-MORE
PROGRAMS"
570 EN$(1)=CHR$(181)&CHR$(19
9)&CHR$(28)&"X-EX BASIC """."
"-MORE PROGRAMS"&CHR$(0)
580 CLOSE #1 :: L=I-1 :: OPE
N #2:"DSK"&DISK$&".CAT",VARI
ABLE 163
590 DISPLAY AT(18,1)BEEP:"RU
N DISPLAY |WRITING"
600 FOR PN=0 TO 4 :: PC=20*P
N :: PL=50*PN :: COL=1
610 ! 1 CALL CLEAR
620 PRINT #2:LN$(1)&CHR$(157
)&CHR$(200)&CHR$(5)&CLS&CHR$
(0)
630 ! 3 DISPLAY AT(1,1):"DIS
K NAME        L PROGRAMS"
640 PRINT #2:LN$(3)&DI$(1)&C
HR$(199)&CHR$(28)&"DISK "&A$
(0)&RPT$(" ",14-LEN(A$(0))-L
EN(STR$(L)))&STR$(L)&" PROGR
AMS"&CHR$(0)
650 ! LOAD 20 PROGRAMS AT A
TIME
660 COL=8 :: FOR I=1 TO 20 :
: IF (I+PC)>L THEN 710
670 ! LN DISPLAY AT(I+2,8):"
A--PROGRAM"
680 DISPLAY AT(20,6)SIZE(-4)
:I+PC :: PRINT #2:LN$(I+3)&D
I$(I+2)&CHR$(199)&CHR$(3+LEN
(A$(I+PC)))&CHR$(I+64)&"--"&
A$(I+PC)&CHR$(0):: NEXT I
690 ! PRESS FOR MORE
700 ! 24 DISPLAY AT(24,1)BEE
P EN$(0) OR EN$(1)
710 PRINT #2:LN$(24)&CHR$(16
2)&CHR$(240)&CHR$(183)&CHR$(
200)&CHR$(2)&"24"&CHR$(179)&
CHR$(200)&CHR$(1)&"1"&CHR$(1
82)&CHR$(238)&EN$((L<21)+1)
720 ! 25 CALL KEY(3,K@,S@)
730 PRINT #2:LN$(25)&CHR$(15
7)&CHR$(200)&CHR$(3)&"KEY"&C
HR$(183)&CHR$(200)&CHR$(1)&"
3"&CHR$(179)&"K@"&CHR$(179)&
"S@"&CHR$(182)&CHR$(0)
740 ! 26 IF S@=0 THEN 25
750 PRINT #2:LN$(26)&CHR$(13
2)&"S@"&CHR$(190)&CHR$(200)&
CHR$(1)&"0"&CHR$(176)&CHR$(2
01)&LN$(25)&CHR$(0)
752 !IF (K@>64)AND(K@<(I+64)
)THEN DISPLAY AT(K@-62,1)SIZ
E(-7):"LOADING"
754 PRINT #2:LO1$(27)&LO2$&L
O3$
760 FOR I=65 TO 20+64 :: IF
(I+PC)>(L+64)THEN 810
770 ! 20 EA IF THEN RUN DSK
PROGRAM
780 ! LN IF K@=65 THEN RUN "
DSK1.PROGRAM"
790 DISPLAY AT(20,1)SIZE(-4)
:I+PC-64 :: PRINT #2:LN$(I-3
7)&IF$(I)&CHR$(0):: NEXT I
800 ! 48 IF K@=88 THEN CALL
CLEAR :: END
810 PRINT #2:LN$(48)&CHR$(13
2)&"K@"&CHR$(190)&CHR$(200)&
CHR$(2)&"88"&CHR$(176)&CHR$(
157)&CHR$(200)&CHR$(5)&CLS&C
HR$(130)&CHR$(139)&CHR$(0)
820 ! 49 IF K@<>32 THEN 25
830 PRINT #2:LN$(49)&CHR$(13
2)&"K@"&CHR$(191)&CHR$(192)&
CHR$(200)&CHR$(2)&"32"&CHR$(
176)&CHR$(201)&LN$(25)&CHR$(
0)
840 IF PN=0 AND L>20 THEN 90
0
850 IF PN=1 AND L>40 THEN 90
0
860 IF PN=2 AND L>60 THEN 90
0
870 IF PN=3 AND L>80 THEN 90
0
880 ! 250 GOTO 1
890 PRINT #2:CHR$(0)&CHR$(25
0)&CHR$(134)&CHR$(201)&CHR$(
0)&CHR$(1)&CHR$(0)
900 IF (I+PC)>(L+64)THEN 930
910 NEXT PN
920 ! END OF PROGRAM LIST
930 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #2 :: DISPLAY A
T(12,21)BEEP:"COMPLETE"
940 DISPLAY AT(24,7):"(Q)UIT
  (R)EDO"
950 CALL KEY(0,K,S):: IF K<8
1 OR K>82 THEN 950
960 IF K=82 THEN 350
970 DISPLAY AT(15,1)ERASE AL
L:"ENTER THE FOLLOWING": :"N
EW": :"MERGE DSK"&DISK$&".CA
T": :"SAVE DSK"&DISK$&".LOAD
"
```

********* *****

* LET TELCO DO THE CALLING *

****

Glenn Bernasek, of TI CHIPS, Cleveland OH, has an article reprinted in the Mid South 99 NL 3/94 in which he describes using the auto-dial feature of TELCO to dial frequently used phone numbers.

"The method is simple. I just load TELCO and choose (A)uto dialer from the main menu. I then press (M) and select the phone line number I want to add or edit. I type in the name and phone number, and leave the modem defaults alone. (Remember to press <FCTN 9> to return to the Main Menu and choose (S)etup options and to save (C)hanges.)

From then on, ...load TELCO, choose (A)uto dialer, PICK UP THE PHONE and enter the phone line number of the person I want to call. Press <FCTN 4> when the FIRST ring sounds. This will isolate me from TELCO and the modem, and I will be able to continue my phone call as I usually would.

You might want to stick "1170," in front of the phone numbers that you don't want `Call Waiting' to interrupt."

```
********************
*    LOADER100     *
* by Jerry Keisler *
********************
```

LITI 99er NL

Writes a load program for a disk full of extended basic and basic programs that run in extended basic.

100 programs max.

```
3 !contains fix so LOADING
will will appear by program
that is loading.

10 !Ever wanted a fast
starting LOAD program for a
disk that was full and you
have no intention of
changing the present
12 !programs. This program
will build a LOAD program
with a selection list of up
to 100 programs that can be
loaded at the touch of one
14 ! key.
16 !THIS PROGRAM WRITES
PROGRAMS!! Due to the nature
of this program, I am
including the explanation in
the program. The new program
18 !is written by sending
the proper ASCII token
values to a DV 163 file that
can be merged into memory,
thus creating a program.
20 !The token values are
sent to a DV 163 file using
CHR$().
22 !All line numbers use two
CHR$()'s. I use 0 in the
first one as my maximum line
number is 250. 255 is the
largest value you can put in
24 !one CHR$(). You must end
each program line with
CHR$(0). The end of program
is CHR$(255)&CHR$(255).
There is no line number or
26 !end of program line in
this line. Everything in a
program line is joined with
the & symbol.
28 !You will note symbols
such as , # ( ) :: etc all
have their own CHR$(). There
are 3 special CHR$()'s.
CHR$(199) indicates a Quoted
30 !string follows.
CHR$(200) indicates an
unquoted string or  number
follows. 199 and 200 must be
followed by the number of
characters in the
32 !string. Example CHR$(199
)&CHR$(10)&"hi you all".
CHR$(201) incicates a line
number follows. It is known
two CHR$()s will follow
this.
34 !Example CHR$(134)&CHR$(2
01)&CHR$(1)&CHR$(3) says
GOTO 258. The CHR$(1)=255
and CHR$(3)=3. You might say
you are dealing with a base
255 number system.
35 !All CALL routines are
handeled as strings. See lin
e 730.
36 !Everything else is
documented in the program.
You can delete all remarks
to speed up the
38 !initialization of the
program. If the program will
not run there is CAT file on
the disk that is not DV 163.
Change CAT in line 580 to
40 !another name and
remember to use it in place
 of CAT when the program end
s.
110 ! A$(0)=DISK NAME
120 ! A$()= PROGRAM NAMES
130 ! B   = FILE TYPE
140 ! C   = FILE SECTORS USE
D
150 ! CL$ = "CLEAR"
160 ! COL = COLUMN #
170 ! D   = FILE RECORD SIZE
180 ! EN$(0)=:"X-EX BASIC"
190 ! EN$(1)=:"X-EX BASIC "
"-MORE PROGRAMS"
200 ! I   = LOOP
210 ! J   = FILE COUNTER 127
 MAX
220 ! K&S = CALL KEY VARIABL
ES
230 ! L   = # OF PROGRAMS
240 ! PC  = PROGRAM BLOCK
250 ! PL  = LINE # BLOCK
260 ! LN  = PAGE #
270 ! Q$  = ACCEPT ALL PROGR
AMS FLAG
280 ! QQ$ = ACCEPT THIS PROG
RAM
290 ! DEF DI$(R)=DISLPAY AT(
R,COL):
300 ! DEF IF$(N)=IF K@="N" T
HEN "RUN DSK1."&A$(I-64+P
310 ! DEF LN$(N)=LINE NUMBER
320 ! NOTE LINE NUMBERS WILL
 CHANGE DEPENDING ON PAGE OF
 PROGRAM SCEEN.
330 CALL CLEAR :: CALL SCREE
N(13):: DISPLAY AT(4,6):"L O
 A D E R  1 0 0"
340 DIM A$(100)
350 DISPLAY AT(6,1):"ACCEPT
ALL PROGRAMS? Y": :"LOAD PRO
GRAMS FROM DISK 1" :: ACCEPT
 AT(6,22)SIZE(-1)VALIDATE("Y
N")BEEP:Q$
352 ACCEPT AT(8,25)SIZE(-1)V
ALIDATE("123")BEEP:DISK$
353 DISPLAY AT(12,1)BEEP:"PR
OGRAM STATUS.......WORKING"
:: CL$="CLEAR"
370 DISPLAY AT(16,5):"DISK N
AME--" :: DISPLAY AT(18,11):
"READING"
380 OPEN #1:"DSK"&DISK$&".",
INPUT ,RELATIVE,INTERNAL
390 ! LINE NUMBER
400 DEF LN$(N)=CHR$(0)&CHR$(
N+PL)
410 ! DISPLAY AT(R,COL):
420 DEF DI$(R)=CHR$(162)&CHR
$(240)&CHR$(183)&CHR$(200)&C
HR$(LEN(STR$(R)))&STR$(R)&CH
R$(179)&CHR$(200)&CHR$(1)&ST
R$(COL)&CHR$(182)&CHR$(181)
422 !IF (K@>64)AND(K@<85)THE
N DISPLAY AT(K@-62,1)SIZE(-7
):"LOADING"
423 DEF LO1$(N)=CHR$(0)&CHR$
(N+PL)&CHR$(132)&CHR$(183)&"
K@"&CHR$(192)&CHR$(200)&CHR$
(2)&"64"&CHR$(182)&CHR$(187)
&CHR$(183)&"K@"&CHR$(191)&CH
R$(200)&CHR$(2)&STR$(I+64)&C
HR$(182)
424 DEF LO2$=CHR$(176)&CHR$(
162)&CHR$(240)&CHR$(183)&"K@
"&CHR$(194)&CHR$(200)&CHR$(2
)&"62"&CHR$(179)&CHR$(200)&C
HR$(1)&"1"&CHR$(182)
425 DEF LO3$=CHR$(235)&CHR$(
183)&CHR$(194)&CHR$(200)&CHR
$(1)&"7"&CHR$(182)&CHR$(181)
&CHR$(199)&CHR$(7)&"LOADING"
&CHR$(0)
430 ! IF K@="N" THEN "RUN DS
K1."&A$(XXX)
440 DEF IF$(N)=CHR$(132)&"K@
"&CHR$(190)&CHR$(200)&CHR$(2
)&STR$(N)&CHR$(176)&CHR$(169
)&CHR$(199)&CHR$(LEN(A$(I-64
```

M.U.N.C.H. INCIME AND EXPENSE FOR THE YEAR ENDING 9-30-94.

INCOME:	
DUES	$321.95
RAFFLE	92.00
ADVENTURE I&II	83.85
D.O.M.	71.00
TONY EALCO CLASSICS	22.00
PICNIC	18.00
NEWSLETTER SUBSCRIPTION	13.00
TOTAL INCOME	$621.80

EXPENSES:	
HALL RENTAL	$300.00
POSTAGE	236.27
DISKS PURCHASED	125.00
NWSLETTER SUPPLIES	118.00
BANK SERVICE CHG.	71.10
PICNIC	49.00
TOTAL EXPENSES	$899.37

CHECKBOOK BLAANCE 10-01-93	$787.29
CHECKBOOK BALANCE 9-30-94	$509.72

RESPECTFULLY SUBMITTED:

JAMES W. COX,
TREASURER
10-01-94

A Short Banner Program

by Tony Falco

There have been numerous banner programs found in this newsletter over the years. So why another one? This one is shorter than most and it allows more options than others that I have seen.

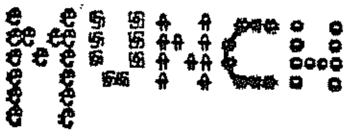
The banner program listed below allows the user to pick ten different character sizes. You simply pick a magnifacation from 1 (8x8 characters) to 10 (80x80 characters) to get the desired size. In addition you can print the banner either horizontally or vertically.

The program was designed for the Gemini 10X printer but it should run on any other printer. You may need to change line 10 for other printers. This line sets the line feed size to 1/12 inch.

This program also shares some disadvantages with other programs. It uses the built in T.I. character set and so with high magnifacation some letters having diagonal lines (N,R and X for example) do not look too pleasing. It also tends to be slow, pausing to convert from hex to decimal to binary (see subprograms in lines 75-95) after each character.

```
10 OPEN #1:"PIO" :: PRINT #1
:CHR$(27);CHR$(65);CHR$(6)::
 CALL CLEAR
15 INPUT "HORIZONTAL/VERTICA
L (H/V)":MD$ :: INPUT "MAGNI
FACATION(1-10)":MG
20 INPUT "MESSAGE==>":M$ ::
FOR X=1 TO LEN(M$):: D=ASC(S
EG$(M$,X,1))
25 CALL CHARPAT(D,P$):: FOR
Y=1 TO 8 :: G$=SEG$(P$,2*Y-1
,1)
30 CALL HEX_BIN(G$,B$):: K$=
SEG$(P$,2*Y,1):: CALL HEX_BI
N(K$,C$)
35 FOR I=1 TO 4 :: A$(Y,I)=S
EG$(B$,I,1):: NEXT I
40 FOR I=5 TO 8 :: A$(Y,I)=S
EG$(C$,I-4,1):: NEXT I :: NE
XT Y
45 PRINT #1:CHR$(27);CHR$(77
);CHR$(40-4*MG):: IF MD$="H"
 THEN 60
```

```
50 FOR Y=1 TO 8 :: FOR J=1 T
O MG :: FOR I=1 TO 8 :: PRIN
T #1:RPT$(A$(Y,I),MG);
55 NEXT I :: PRINT #1 :: NEX
T J :: NEXT Y :: GOTO 70
60 FOR Y=1 TO 8 :: FOR J=1 T
O MG :: FOR I=1 TO 8 :: PRIN
T #1:RPT$(A$(9-I,Y),MG);
65 NEXT I :: PRINT #1 :: NEX
T J :: NEXT Y
70 NEXT X :: CLOSE #1
75 SUB DEC(HX$,DD):: V=ASC(H
X$):: DD=(V-48)*(V>47)*(V<58
)+(55-V)*(V>64):: SUBEND
80 SUB BINARY(DD,BD$):: BD$=
"" :: FOR X=3 TO 0 STEP -1
85 IF DD>=2^X THEN DG$=CHR$(
79):: DD=DD-2^X ELSE DG$=" "
90 BD$=BD$&DG$ :: NEXT X ::
SUBEND
95 SUB HEX_BIN(HX$,B$):: CAL
L DEC(HX$,D):: CALL BINARY(D
,B$):: SUBEND
```



Short Calendar Programs

by Tony Falco

On what day of the week was the Declaration of Indepeendence signed? On which day does Christmas fall next year? On what day of the week did the Blizzard of 78 strike? On which day of the week were you born? Your birthday for next year falls on which day? What about your 100th birthday, or Mom and Dad's 5oth aniversary, or New Year's day in the year 2000?. You could look these dates up in a perpetual calendar found in some almanacs or easier still you can enter the first two line program shown below. To run the program simply enter the month, day and year (separated by commas) for the desired date when propmpted. For example: 7,4,1776 for the day that the Declaration of Independence was signed. (The program uses an algorithm called the Zeller Congruence Formula.)

The date on which Easter Sunday falls is not simple to figure out without the use of astronomical tables and almanacs. In 325 A.D., it was decided that the date would be set according to the Gregorian calendar. Easter was to be the first Sunday following the full moon on or after the vernal equinox. (about March 21.) Easter can fall, therefore, on any Sunday between March 22 and April 25. Christians in Western Europe and the Americas still use this method. Since the date is determined by two periodic events, the full moon and the vernal equinox, it should be possible to devise a numerical scheme for finding the date. The second program listed below does just that. To use the program you just enter the year for which you want the date. The program will print a screenful (Why waste a good algorithm on just one year?) of twenty dates for Easter centered about the desired year.

```
1 INPUT "M,D,Y ==>":M,D,Y$ :: M=M+2*(M>2
)-10*(M<3):: A=VAL(SEG$(Y$,1,2)):: B=VAL
(SEG$(Y$,3,2))+(M>10):: F=INT(2.6*M-.2)+
D+B+INT(B/4)+INT(A/4)-2*A
2 F=F-7*INT(F/7)+1 :: PRINT SEG$("SUNDAY
   MONDAY   TUESDAY  WEDNESDAYTHURSDAY F
RIDAY   SATURDAY",9*F-8,9):;:;:: GOTO 1
```

```
1 INPUT "YEAR=>":X :: FOR Y=X-9 TO X+10
:: D=19*Y-361*INT(Y/19)+24 :: D=D-30*INT
(D/30):: E=6*Y-8*INT(Y/4)-28*INT(Y/7)+6*
D+5 :: E=E-7*INT(E/7)
2 PRINT "In";Y;"Easter is ";SEG$("MarchA
pril",-(D+E<10)-6*(D+E>9),5);D+E+22+31*(
D+E>9);CHR$(32-28*(X=Y)):: NEXT Y
```

MUNCH

BALLISTICS SIMULATION BY SPRITES

by Tony Falco

"I shot an arrow into the air and where fell I know not where." The branch of physics which answers this question of just where an object will land is ballistics. With the program listed here you shoot your own arrows (bullets, missiles, grenades, baseballs, or whatever your imagination dictates) into the air and find out where they will land. You have more control than an artillery man. He has shells with specific velocities and can vary the angle at which the shell exits his gun. You can pick both the velocity and the angle but you will get more of a feel for the real situation if keep the velocity fixed and vary the angle. You will soon discover that the maximum range for a given velocity occurs at a 45 degree angle.

The object of the program is to pick an angle to hit a randomly placed target. You have the choice seeing the path the object follows or simply seeing the object in motion. When you hit the target the screen border will flash red and then a new target will appear.

```
10 RANDOMIZE :: CALL CLEAR :: !BALLISTIC
S SIMULATION BY SPRITES
20 DISPLAY AT(4,2):"SHOW PATH (Y/N) Y" :
: ACCEPT AT(4,18)SIZE(-1):@$
30 CALL CLEAR :: L=45 :: VV=55
40 FOR N=1 TO 14 :: CALL COLOR(N,2,15)::
 NEXT N :: RR=INT(20*RND)+10
50 FOR A=1 TO 28 :: CALL SPRITE(#A,42,2,
256,1):: NEXT A :: CALL SCREEN(6)
60 CALL VCHAR(1,3,124,24):: CALL HCHAR(2
4,1,95,32):: CALL HCHAR(24,RR+1,31)
70 DISPLAY AT(4,2):"TARGET=";RR ::
80 DISPLAY AT(1,2):"ANGLE (11-86)=>";L :
: ACCEPT AT(1,18)SIZE(-2):L
90 DISPLAY AT(2,2):"SPEED (10-70)=>";VV
:: ACCEPT AT(2,18)SIZE(-3):VV
100 IF VV>70 THEN 90 ELSE V=VV/10
110 A=PI*L/180 :: R=V*V*SIN(2*A):: COSA=
COS(A):: TANA=TAN(A)
120 DISPLAY AT(3,2):"RANGE=";R :: X=0 ::
 CL=(ABS(RR-R)<=1)
130 FOR XX=0 TO R-.1 STEP R/28 :: X=X+1
:: Y=(-1/(2*V*V*COSA^2))*(XX^2)+XX*TANA
140 IF @$="Y" THEN J=X ELSE J=1
150 IF Y<=24 AND Y>0 AND XX<=30 THEN CAL
L LOCATE(#J,192-8*Y,8*XX+12)
160 NEXT XX :: CALL SOUND(-100,220,0,-7,
0):: IF CL THEN CALL SCREEN(7):: GOTO 30
170 FOR D=1 TO 500 :: NEXT D :: CALL CLE
AR :: GOTO 50
```