THE CUGS

MONITOR

CUGS ◄

CUGS ◄

SUMMER ● 1987!

IN THIS ISSUE

EDITORIAL

Summer's gone - kid's goin' back to school - yard's
turning yellow and filling with leaves - garden
turning black from frost - stores are setting out
HALLOWE'EN CANDY and costumes (!!!) ... AND THE
"CRAZIES" AT CUGS ARE BACK AT IT AGAIN!!! - Yep ...
it's fall, alright!

This issue (coming to you courtesy your executive
and Canada Post WE HOPE!) we'll keep it "light" and
you can consider this a combined EDITORIAL/CRYSTAL
BALL article. Next month, I'll get intense once
more.

But seriously folks, I hope you have enough summer
memories to warm your winter, and I hope you're as
anxious as I to get back into hobby computing (at
least, that's what I call it - my wife calls it an
obsession). Actually, like many of your machines,
my computer didn't get the summer off. Especially
as an executive member, I've been working with the
others on our "revamp" of the club DISK LIBRARY.
While we're a little behind schedule (we learned
that phrase from COMMODORE CANADA!), we've got a
respectable number of disks done and listed. Done
means SORTED, CATALOGED, CHECKED FOR OPERATION,
ANNOTATED AND PLACED IN A COLLECTION WITH A LOADER
PROGRAM!

Our fall planning this year will centre in on
COMMUNICATIONS. In a short survey of some of our
club faithful over the last couple of spring
meetings, we determined that many were INTERESTED
in computers for COMMUNICATION, but few had become
involved mostly due to too little information.
Well, we're gonna fix all that! Starting with the
September meeting, we intend to make some topic on
communication a regular part of each meeting's
presentation. We'll continue until our members
know it all or feel they've learned enough!

We hope to feature a COMMUNICATIONS article in each
issue of the MONITOR to complement the
presentations, as well as the columns you've come (
WILL come, to new members) to know and love! (e.g.
the EDITORIAL).

TUNE IN NEXT MONTH FOR "DAVID VS GOLIATH"
(SUBTITLE: THE C64/128 VS THE "KILLER-BYTE"
MACHINES!)

zzzzZIPPERSssss

Housecleaning for Your Computer

by HARRY CHONG

Computers get ornery for many reasons, although
electrical jolts, heat and dirt are the principal
causes. Following are some suggestions on how to
stay on top of these gremlins:

    -Set your computer up in a clean, dry,
    well-lighted area. Damp, dusty, smoky, hot or
    greasy atmospheres are absolute NO-NO's.

    -A particle of cigarette ash can act like a
    boulder to a disk drive head so, butt out.

    -Crumbs and liquids spilled on or into a
    keyboard, disks or tapes are disastrous.

    -Beware of GREASY hands, STICKY fingers - who
    needs 'em? Your COMPUTER don't - fer sure.

    -OFF LIMITS to pets. Cat and dog hairs get
    into keyboards, disk drives and disk
    envelopes, causing lots of mischief.

    -Periodically vacuum the outer surfaces of
    your equipment.

    -If you dust, use a cotton cloth - they do not
    generate static electricity.

    -Never use an 'IONIZER' near your computer.
    This device causes your system to attract the
    dust that normally floats aimlessly around the
    room. Anything that increases static should
    be avoided.

    -Just remembered something I read - can't
    quite put my finger on the source. If you do
    your computing off and on through the day, and
    if you have a fan keeping your equipment cool,
    then it's best to leave your system on.
    Turning it off and on too many times could put
    it into electrical shock. Some people
    virtually never turn their systems off. One
    BIG exception: if an electrical storm is on
    the horizon, turn everything off and pull the
    MASTER PLUG!

A CAUTIONARY NOTE: BRIGHTNESS KILLS! Keep your
brightness control turned down when you're not
computing. Overly bright displays are bad for the
EYES, and shorten screen life.

SIR RICHARD'S BASIC

by RICHARD MAZE

Last month I looked at one-dimension arrays. I examined what an array is and some methods of putting data into an array. In this article I want to explore two-dimension arrays and some further uses of arrays in programs.

If we visualize an array as a shelving unit, then a one-dimension array is simply a vertical set of shelving, each shelf having the same variable name with a number representing its position in the unit. For example, an array NA$ could have 21 shelves numbered from 0 to 20. Refering to a certain shelf would simply mean refering to the NAME of the array followed by the NUMBER of the shelves (in parentheses). NA$(14) refers to the data stored on SHELF NUMBER 14 of the UNIT called NA$.

A one-dimension array has a "VERTICAL" component only. This consists of the number of shelves. A TWO-dimension array has BOTH A VERTICAL AND HORIZONTAL component. Think of it as a shelving unit with several shelving units standing side-by-side. Such an array is treated exactly like the one-dimension arrays except BOTH dimensions must be expressed. For example: to store a number of names, streets, cities and provinces in ONE array, the VERTICAL component for this array might be the NUMBER of PEOPLE we want (say 20). The HORIZONTAL component would be a SET OF SHELVES EACH for NAME, ADDRESS, CITY and PROVINCE (total of 4). If we used the array name DA$ to hold this data then we set up the array using the DIM statement as: DIM DA$(20,4). Note that the vertical component is first and the horizontal component second. A comma separates the two components. (NOTE: the array actually has 21 vertical and 5 horizontal shelves because a shelf 0 exists in each direction. Usually, the 0 shelf is not used because it can create counting confusion). A particular set of data can be retrieved by refering to its position in the unit. For example, the CITY for the 13th person would be DA$(13,3), the NAME of the 16th person is DA$(16,1). Remember, the data must be all of one TYPE (string or numeric). The TYPE of data to be stored in the array is set by the DIM statement. You cannot mix string and numeric variables in the same arrray.

TWO-dimension arrays are most often referenced using NESTED loops (loops within loops). If you have data to put into the array and the data is in the order: NAME, ADDRESS, CITY, PROVINCE, you could fill the array using the following program segment:

```
100 REM FILL ARRAY DA$ USING READ
110 DIM DA$(20,4):REM 1 ARRAY WITH 21 X 5 SHELVES
120 FOR I = 1 TO 20:REM IGNORE SHELF 0
130 :    FOR J = 1 TO 4:REM IGNORE 0 HERE TOO
140 :        READ DA$(I,J)
150 :    NEXT J
160 NEXT I
```

If the data was in the order: all NAMES, then all ADDRESSES, then CITIES, then all PROVINCES you simply change the order of the loops so the program becomes:

```
100 REM FILL ARRAY DA$ USING READ
110 DIM DA$(20,4):REM 1 ARRAY WITH 21 X 5 SHELVES
120 FOR J = 1 TO 4:REM IGNORE SHELF 0
130 :    FOR I = 1 TO 20:REM IGNORE 0 HERE TOO
140 :        READ DA$(I,J)
150 :    NEXT I
160 NEXT J
```

Like one-dimension arrays, two-dimension arrays can be filled using the results of calculations, INPUT, GET or READing files off disk. The main advantage of TWO-dimension arrays over one-dimension arrays is that they allow storing large amounts of RELATED data using only one variable.

Arrays are often used in the following situations within a program:

- with large amounts of related data
- adventure games (rooms and contents)
- databases (address files, etc.)
- calculations (sales records, etc.)
- disk catalogs
- whenever data is to be sorted
- when data is brought in from a disk file

The major advantage of using an array is the control of variables a programmer has. It is much easier to keep track of one or two array variables than the number of variables needed without the array. Another advantage is that programs are often shorter because data is accessed and used in short loops - impossible with a large number of separate variables. A third (and probably most important) advantage is the large quantity of data that can be stored and made easily available in an array.
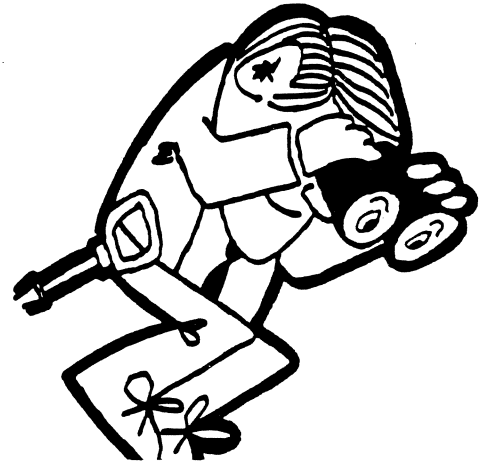
ML from Ed

THE SCROLL OF THE WILD (ED, THAT IS)

by ED DIETRICH

LINE SCROLL WEDGE

This program is similar to last month's. It will scroll the top line of the screen from right to left. One difference is that it moveS color memory along with screen memory so that different colored text scrolls. Another change is that this program is 'wedged' into the TIMER INTERRUPT of the computer's operating system. This means that the computer handles the line scrolling AUTOMATICALLY no matter what kind of program is running or whether one is running at all. The only drawback to this technique is that it creates timing irregularities during disk access. Because of this, it is wise to switch off the scrolling before accessing the drive.

To enable scrolling type SYS 49152 and press RETURN.

To disable scrolling type SYS 49170 and press RETURN.

To change the scroll speed POKE 49189 with any number from 1-255. The higher the number, the slower the scroll speed. Change speed only when the scrolling is not active.

To enter the program use XMON64C(50135) from our Club PROGRAMMING Library Disk. Ignore the line numbers. They're for reference only.

The program starts at $C000 (49152 decimal). The first line entered should be;

    .A C000 SEI

Type in the rest of the program following the prompts.

1.	SEI	16.	RTS	31.	LDA	$C065
2.	LDA #$14	17.	DEC $C064	32.	STA	$0400,X
3.	STA $C064	18.	BNE $C061	33.	LDX	#$00
4.	LDA #$1F	19.	LDA #$14	34.	LDY	#$01
5.	STA $0314	20.	STA $C064	35.	LDA	$D800
6.	LDA #$C0	21.	LDX #$00	36.	STA	$C065
7.	STA $0315	22.	LDY #$01	37.	LDA	$D800,Y
8.	CLI	23.	LDA $0400	38.	STA	$D800,X
9.	RTS	24.	STA $C065	39.	INY	
10.	SEI	25.	LDA $0400,Y	40.	INX	
11.	LDA #$31	26.	STA $0400,X	41.	CPY	#$28
12.	STA $0314	27.	INY	42.	BCC	$C04F
13.	LDA #$EA	28.	INX	43.	LDA	$C065
14.	STA $0315	29.	CPY #$28	44.	STA	$D800,X
15.	CLI	30.	BCC $C033	45.	JMP	$EA31

Notable Memory Locations:

$0314 & $0315 - HARDWARE INTERRUPT VECTOR
            ($EA31).
$0400 - Beginning of SCREEN MEMORY in RAM.
$D800 - Beginning of COLOR MEMORY in RAM.
$EA31 - Beginning of Commodore 64 TIMER INTERRUPT
handling routine (house-keeping tasks).

A Brief EXplanation:

Lines 1-9 enable the 'interrupt wedge'.
Lines 10-16 disable the wedge and return interrupt handling to normal.
Lines 17 & 18 are a timing loop to slow down the scrolling speed by a factor of twenty. As the interrupt occurs sixty times per second, the scroll speed would otherwise be sixty characters per second (pretty, but illegible).
Lines 18-44 form our 'wedge' program.
Lines 19 & 20 reload our timer with a value of twenty.
Lines 21 & 22 prepare the X & Y registers to be used for indexed adressing to move the contents of the top line of the screen left one character.
Lines 23 & 24 copy the character in the top left hand corner of the screen to a temporary memory location.
Lines 25-30 are a loop which move the top line characters to the left one by one.
Lines 31 & 32 copy the contents of the temporary memory location to the top right corner of the screen enabling the 'wrap-around' feature of the scroll.
Lines 33-44 are identical to lines 21-32 except that they move the top line of the color RAM as opposed to the screen RAM.
Line 45 jumps to the beginning of the normal timer interrupt routine which performs the computer's house-keeping tasks and returns control to the main program (if one is running).

This program, once entered, can be saved from within XMON with the command;

    .S "0:filename",08,C000,C064
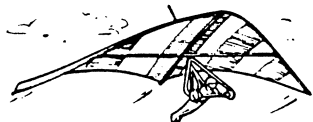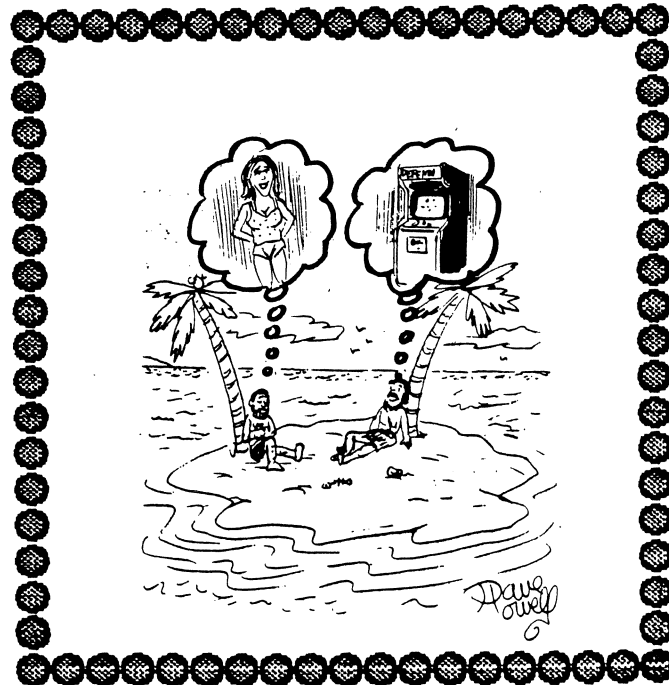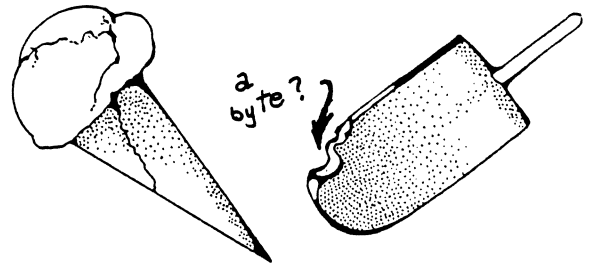
Disk-Etiquette

CUGS DISK LIBRARY NOTES

by Earl Brown

Well, how are you all? The summer is over and I do hope you had a good one. Personally it was much too short for me.

The disk library has not had a complete "overhaul", but you can tell by the new catalogue, we managed to get quite a lot done. By next month, I hope the transition will be complete for the 64 programs, as well as several disks prepared for 128 owners. We have the programs, it's just a matter of checking them out and "ID"-ing them into CUGS own library. When you purchase most new disks from our 64 library, you'll notice the first program is entitled "UGS LOADER". This fine menu driven loader, was written by our club president, Richard Maze in BASIC. It will soon have a customizing ML routine added, being written by our former president and ML guru, Ed Dietrich. By listing the program, any member should be able to include a scrolling message of his/her own choosing as well as a screen color change for your own personal use.

The loader program also loads a sequential file entitled "CUGS DATA". If you load this file into a word processor (seq), you can see how it's made up, and be able to make your own with the same word processor. Richard is planning to write a basic program for those who would rather use a prompt routine for creating this file instead of a word processor. This program will be included on a future disk. A word of warning regarding this CUGS LOADER program should be mentioned. Every program it loads, including BASIC ones, uses the ",8,1". So, if the loader program fails to load any BASIC program properly identified in the seq. file "CUGS DATA", that program may not be located at the usual C-64'S start of BASIC (there are different reasons why this might be). In that case, your best bet would be: (1) load that particular program manually using the standard ",8" routine - (2) make sure you've loaded it, (3) scratch the original program from the disk, and (4) save the program again under the same name (using ",8"). The CUGS LOADER should now load the file.

Use this shortcut to load and run the LOADER:

        10"",8: (shifted run/stop)

We appreciate your comments on our new (and improved?) library.

MEETING PLACE

AGENDA:

WELCOME ADDRESS - PRES. RICHARD MAZE

PRESENTING THE "NEW IMPROVED" CUGS DISK LIBRARY!

**** Coffee *** Visits *** Library Checking ****

IF A BAUD-Y MEET A BAUD-Y DOWNLOADING ON THE LINE!

(AN HONEST-TO-GOODNES, REAL B E G I N N E R S INTRODUCTION TO TELECOMMUNICATING.)

Bulletin Board List

BBS Name	Phone	Hours	baud rate
Abyss	584-0721	24 hrs.	300
Bit Bucket BBS	352-3236	6pm-8am, Sun.	300/1200
DUH Systems BBS	545-6953	24 hrs.	
DataPac 1200	565-0181	24 hrs.	1200
DataPac 2400	565-6000	24 hrs.	2400
DataPac 300	565-0111	24 hrs.	300
Digital Dimensions.	586-6493	24 hrs.	300/1200
Genesis	345-2265	6pm - 7am	3/12/2400
Micro City 1	584-0747	24 hrs.	300/1200
Micro City 2	584-0748	24 hrs.	300/1200
Mouse Trax (Apple)	359-7055	5pm - 8am	300/1200
Negative Zone	565-8538	24 hrs.	300/1200
Onyx FOG # 50 RCP/M	949-1564	24 hrs.	3/12/2400
Orion	789-9754	24 hrs.	300/1200
Parkland Opus	543-9159	24 hrs.	3/12/2400
Polestar Opus	586-1551	24 hrs.	3/12/2400
Probable Fate System	525-1054	24 hrs.	300
R.A.T. 2	949-6105	24 hrs.	300/1200
Regina Fido	347-4493	24 hrs.	3/12/2400
Sask-Ed-Net 1	525-3973	24 hrs.	300/1200
Sask-Ed-Net 2	522-1998	24 hrs.	300/1200
Shadowland	789-7883	6pm-8am, Sun	300/1200
Tesseract	757-5699	24 hrs.	300/1200
Trip. Diam. CompuMag	775-3314	24 hrs.	300
Turbo BBS	586-7560	24 hrs.	300/1200
Ultra Fast Systems	545-5717	24 hrs.	300/1200
Unibase 1 (E.M.I.S)	584-2035	24 hrs.	300/1200
Unibase 2 (E.M.I.S.)	586-5585	24 hrs.	300/1200

G. R. PHONE HOME!

by GREG REZANSOFF

Over the next few months, Cugs will be presenting a
series of interesting and informative talks on
computer communication. A small survey made it
quite clear to see that many club members have some
interest in the topic.

Much of the information will be intended for the
new user, starting with the very basic concepts,
and moving to the more technical aspects. Along
with this, I have planned to do software reviews of
several different terminal packages in the MONITOR.

The terminal program is the single most important
piece of software in computer communication, and
deserves our attention. These reviews will
describeption each program and rate it on several
characteristics. This will hopefully allow you to
compare and decide which best suits your needs.

TERMINAL PACKAGE review #1:   "CBTERM"

Originally created by the people at COMPUSERVE,
CBTERM64 is a versatile and powerful piece of
software which takes full advantage of the power of
a modem. Not only is it versatile, but its price
is terrific. CBTERM is in the public domain, which
means it is FREE for the taking. This may be
important if you are concerned with keeping
expenses to minimum.

CBTERM64, like most terminals, is eqiupped to
handle parity, baud rate, manual and auto dialing,
and a few other things. Although it does not have
a "telephone book" like many commercial packages,
the options available will suffice in most
situations.

As you use the program, you'll find that many of
its commands are for handling files, or incoming
information. It has a large (22k to be exact)
buffer size that can be edited, saved, or
transmitted over the modem. It not only handles
ASCII files, but has x-modem protocol, for

uploading and downloading programs.
Regarding visual output and graphics, CBTERM has a
few features which aid in viewing and typing. Both
40 column and 80 column modes are supported, so
information will be properly formatted on the
screen. In 80 column mode, a command line is
displayed on the bottom of the screen allowing you
to type data in before it is sent (allowing for
quicker and smoother access).

With all these features, it is easy to see that
CBTERM64 can be a useful piece of software for the
beginner, or for someone who needs a "rough and
ready" termial program. In either case, there is
no reason not to have a copy of it - it is public
domain, and free to anyone who wants it.

It is listed in the CUGS library along with several
others, or you can get a copy of it from me, or
from several bulletin boards here in Regina.

WHAT I DID FOR

←LOVE→

('CAUSE THEY SURE DON'T PAY ME!)

by Steve Bogues

(Editor's note: We intended to print a paragraph
from each club executive member describing his/her
job for the interest of new and old members, but
only Steve was brave enough to respond [so far!].
Election-time's just two months off, and some of
you should consider serving a year or two. Read
what Steve has to say!)

A few weeks ago Ken asked me to try and define what
a "member at large" (Steve is a CUGS MEMBER AT
LARGE) really was, and what his duties were. I was
to touch on any high points during my tenure.

The best definition I could come up with is - a
MEMBER AT LARGE is a person that can assist the
other executive members in the completion of their
proper duties in the club.

When I joined the executive as a MEMBER AT LARGE, I
felt a bit like the "new kid on the block". I had
a little BASIC training and very little experience
with machine language. I certainly began to wonder
what i could do to help the club.

This feeling passed after the first executive
meeting or two. I soon began to learn the ropes of
the club's operation by watching the senior
executive members at work. I must admit, my
thoughts of long hours of dull work and thankless
dedication were false.
                                        The
executive was so well organized that all the club
functions were split equally and fairly according
to the talents of the member. i found that the
jobs I was assigned were interesting and turned out
to be a lot of fun, too. In fact, I would say that
the chance i had to explore the old club disk
library for the new led me to find many interesting
programs and utilities that I now use on a regular
basis.

in the past year, I have found that my knowledge of
the C64 has increased greatly thanks to the help
and assistance of the senior executive. There have
been several occasions that they have helped me
through a tough hardware or software problem.

I can safely say that the club has made great
strides forward after the unexpected departure of
many long-time members to the AMIGA USERS' GROUP.
The club paper, the MONITOR, has been
re-established, the DISK LIBRARY is undergoing
revision, we had a super display at the first
COMPUTERFEST last February, and club membership is
on the rise!

In short, I personally feel that anyone who would
like to get a bit more involved in the club would
find a year or two on the executive well worth the
small effort, and a lot of fun to boot!