

C. U. G. S.

Manufacturer

MERRY CHRISTMAS



FROM C. U. G. S.

W. A. H.

D. C. E. Y.

OBLIGATORY STUFF

PRESIDENT
VICE-PRESIDENT
TREASURER
LIBRARIAN
ASS'T LIBRARIAN
EDITOR
ASS'T EDITOR
MEMBERS AT LARGE

Richard Maze
Barry Bircher
Harry Chong
Earl Brown
Steve Bogues
Ken Danylczuk
Greg Rezansoff
G. Williams



EDITORIAL

THE JOY OF LEX!

THE MONITOR is published monthly by the COMMODORE USERS' GROUP OF SASKATCHEWAN (CUGS), Regina, Sask., Canada. CUGS meetings are held at 7 pm on the first Wednesday of every month (unless otherwise noted) in the North-West Leisure Centre, corner of Rochdale Boulevard and Arnason Street.

Anyone interested in computing, especially on the C64, 128 or 64C, is welcome to attend any meeting. Out of town members are welcome, but may be charged a small mailing fee for newsletters. Members are welcome to submit public domain software for inclusion in the CUGS DISK LIBRARY. These programs are made available to members. Any member is entitled to purchase DISKS from our public domain library for a nominal fee. Some programs are 'freeware', some are from computer magazines, or the public domain. Individual members are responsible for deleting any program that he/she is not entitled to by law (you must be the owner of the magazine in which a particular program was printed). To the best of our knowledge, all such programs are identified in their basic listings. Please inform us otherwise.

CUGS is a non-profit organization comprised of C64, 64C, C128, and 128D users interested in sharing ideas, programs, knowledge, problems and their solutions with each of its members. The more members that participate, the better the variety of benefits. Membership dues are pro-rated, based on a January to December year.



IN THIS ISSUE:

- DECEMBER REMEMBERING - Maze's memories!
- MEETING PLACE - Date, Time, Place, Agenda
- EDITORIAL - The JOY OF LEX!
- SIR RICHARD'S BASIC - More on diskfiling!
- SCRATHC 'N' SAVE - EARL'S LATEST COLUMN!
- A STAR IS BORN! - A hardware review!
- CHRISTMAS ASSEMBLING - A ML assembler review.
- MY AMI VICE - First byte by Bill Bircher
- UNCLE CUGS WANTS YOU! - Membership dues due!

MEMBERSHIP NOTE

Memberships for 1988 now available

Cost: \$10.00 (Jan. to Dec.)

Benefits: Access to Disk Library

Funds used to update disk library, purchase club hardware and print the newsletter!

7% discount at Software Supermarket!

See the treasurer or an exec. member!

"'Tis the season to be jolly", and I am! I've reflected on my past year (only 1!) as editor of this noble rag, and I have to admit the sin of self-pride. Actually, it's not really SELF-pride, it's pride in our association. Anyone who keeps back issues might re-read the October Issue (1986), when I began as editor. You'll see a small paper with little more than editorial comment! This was the year of the great BEG! I felt like each issue was put together by material EXTRACTED, under great duress, from poor overworked executive members.

But dreams do come true!

I really sincerely debated scrapping this editorial, because I don't want CUGS to become complacent. Then, the Christmas Spirit hit, and I figured - what the H..L! It's Christmas! Give 'em a break! So I will!

Seriously, folks. This issue should be a testament to the vitality of our club. As editor this issue, my primary concern has been gathering, proofreading and paste-up (as it should be). Short of this VOLUNTARY article (which many suggest ought to BE forgotten), this issue consists of the literary efforts of SEVEN CLUB MEMBERS! That's TERRIFIC!!!

All of which brings me to my short commentary on the 'JOY OF LEX' (as in lexicography - the study of words). This issue is an example of a joy, a paper truly representing the efforts of many. This coming year seems to be shaping itself up in the computing software industry as the 'YEAR OF THE DESKTOP PUBLISHER' - virtually every software house and its mother are producing and promoting THE program to suit all your publishing needs. And these programs will sell well, because everybody wants their documents to look sharp!

Unfortunately, what is missing from these packages is a WARNING for the purchaser something like "PURCHASE OF THIS ITEM IN NO WAY ASSURES YOU OF PRODUCING AND INTERESTING, ATTRACTIVE, INTELLIGENT OR IMPRESSIVE DOCUMENT" or "CREATIVITY, INSIGHT AND INTELLIGENCE NOT INCLUDED". You can tell me I blew it this time next year, but I suggest that the Desktop flurry will diminish rapidly when those purchasers realize that the neat alignments and canned graphics don't make a good publication - there is (as yet) NO COMPUTER SUBSTITUTE intelligent editing, creative writing and careful human attention in the development of an interesting readable document.

Sure, the programs make the mechanics easier, but at what price? I've had people look at a copy of the MONITOR and ask "What did you use to create the graphics and text?" The answer they want is "SUPER-DUPER DESKTOPPER", but the honest answer is SEVERAL desktop-style programs, a collection of graphic artwork, my own and several others special graphics! In short, a lot of careful planning and hard work! NO ONE PROGRAM CAN (or probably ever will) provide everything one might need!

The editor of your local club newsletter, paper, mailer or what-have-you doesn't need anything more than a decent word-processor, and YOU! NOTA BENE: (that's Latin) One 200 word article, review or commentary from each member of our club would FILL our newsletters for the next TWO YEARS! (without editorials!)

A MERRY COMPUTER CHRISTMAS TO ALL OF YOU!

MEMORIES!

a Message from the President!

December marks the end of another year for CUGS. I would like to take this opportunity to review the past year and take a look forward to next year. In 1987 our meetings were structured a little differently than in the past. Throughout the year, short presentations were made on BASIC statements by members of the executive. We also had a series of presentations on machine language by Ed Dietrich. The last half of each meeting was a major presentation on a program or related group of programs, which was conducted by a club member. These presentations included: GEOS, a Music Software Survey, Paperclip, Income Tax, ChartPak, Multiplan, Communications, Xmodem terminal program, Disk Log, and a hands-on software survey. I would like to take this opportunity to thank everyone who gave a presentation. They were all well-done and much appreciated.

Another change this year was the resurgence of our newsletter, The CUGS MONITOR. Under the capable hands of editor Ken Danylczuk, our newsletter became an integral part of our club. Thank you, Ken, for all the time and effort you have put into making the newsletter successful. I would be remiss if I didn't also thank everyone who contributed to the newsletter by writing articles and reviews.

A major project by the executive was the reorganization of the club's disk library. This very important project was under the direction of librarian Earl Brown. Now, a club member can get a disk which only contains programs concerning one topic. I would also like to thank all the members who have contributed programs to our library.

In February, the Apple Users group sponsored 'Computerfest'. Our club set up a display which was successful in attracting new members to our club. Thanks to everyone who contributed time to help work at the display.

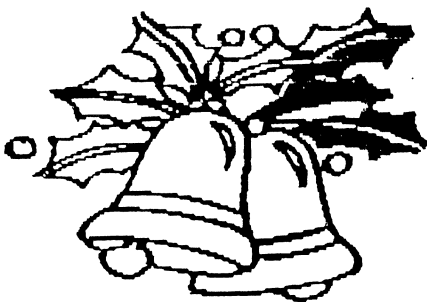
I would also like to thank the members of the executive who gave their time and effort to make my job so easy this year. To those members of the executive not returning this year, thank you for your efforts this past year. I wish you well in your future computing endeavors. I would like to thank the executive members, new and returning, for the coming year. I look forward to working with you for another year.

Looking forward to 1988, the executive have not firmed up all plans at this time, but, there are some ideas that have been discussed. One thing is to again participate in 'COMPUTERFEST'. Although details are sketchy at this time, 'COMPUTERFEST' will be in February again, and I have indicated our club's interest in participating. More details will be forthcoming.

Your executive are also looking at the possibility of having a PUBLIC DOMAIN SOFTWARE SWAP NIGHT for members. Further aspects of 1988 plans will be outlined as soon as they are firmed up. If you have any ideas for presentations or activities, please let an executive member know as soon as possible.

A club is only as strong and viable as its members make it. It requires the involvement of each member in any aspect he/she can. Contribute articles for the newsletter, contribute programs for inclusion in the disk library, volunteer to give presentations, or give us ideas of programs you'd like to see presented, and, most importantly, contribute by attending the meetings and sharing your ideas and knowledge with others.

Richard Maze



A Star is Born!

A NEW 'STAR' IS BORN - THE STAR NP-10!

In the last few months you may have noticed computer shop fliers and advertisements promoting the STAR NP-10. This little printer seems to be getting quite popular with many computer owners, including me!

The NP-10 is very similar to its big brother, the NX-10. Some of the features of this printer are its simple and appealing design, its ease of use and maintenance, and front-mounted control panel allowing line on/off, line/form feed, letter-quality/draft mode select. The front panel controls select HEX dump, forward and reverse microfeed. The printer allows tractor and friction feed, and has automatic paper loading. It can be set to emulate the MPS-801, with all Commodore graphics, and can be set to print NLQ, Italic, Elite, Pica, Super- and Sub- script, Variable Pitch, Bold-face, Expanded, Emphasized, Condensed and Proportional fonts. Several fonts can be mixed on one line.

The printer uses a standard interface and can be used with any other standard interface computer. It has programmable line spacing, uni- or bi-directional printing, ten different language character fonts, and has a programmable ram for downloaded character set.

In general, the manual is very easy to follow and understand. The printer (so far) seems reliable and rugged. With a 1 year warranty, numerous repair centres in Canada, and a Canadian price tag of about \$275.00, I would strongly recommend this printer to anyone in the market, from student to small business user!

Steve Bogues

MEETING PLACE: AGENDA

President's Report - R. Maze



The remainder of the meeting will be devoted to reviews and demonstrations of some of the BEST software and hardware available for the C64/128.

All items supplied for the evening courtesy Bart Ritchie of SOFTWARE SUPERMARKET! A club "Thank you" to BART!

128 ML PROGRAMMING

With the L.A.D.S. ASSEMBLER

by Barry Bircher

Well Ken... you asked for it, so now you're gettin' it! A review (of sorts?). I have never written one, much less used a word processor, so here goes!

First, a little about my obsession ... ahhh ... MYSELF. I am a relatively new computerist/programmer who got "bydden" by the bug two years ago when I saw an ad in the paper for a VIC 20. I know now the store was selling off all it's stock for a measly \$49.99 (plus tax). For a guy who spent \$119.95 (in 1974 dollars) for a new-fangled calculator able to do division and multiplication in mere microseconds (it had 4 functions plus square root!), this price seemed too good to be true - a whole keyboard and 20k (3.5k ram) memory able to do calculations I'd never heard of! I had always wanted to learn how these little devils ticked, so off I went with \$75.00 in my pocket and my heart set on getting one, only to find out they were sold out! The next best thing was a C-64 for \$199.98! Of course, you need a disk drive (\$325.98), joystick, printer, etc., and..... oh Ya!... a program to impress your wife to show her, without a doubt, that we ABSOLUTELY CAN'T LIVE WITHOUT A COMPUTER (and to help making any sense out of my VISA statement soon to follow).

After several months trying to figure out the Users' manual and buying other books and magazines, I soon felt pretty comfortable in BASIC and the 64 in general.

However, I found BASIC was sssslloooooowww. Of course, this is because it's an INTERPRETED language. Machine language was the way to go if speed was important. ML, being foreign, sort of scared me when I looked at a DISASSEMBLER's listing. Egyptian hieroglyphics would seem easier to learn. After several more books on the subject and many hours of wife ignorin.....er... reading, I attempted my first program. It turned out OK, after 1.5 hours of programming and 1.5 hours of debugging. All it did was poke a page of memory to the screen so that the user could see inside the computer. Sounds super simple in BASIC, but in ML, a major accomplishment for me! The program was only about 20 bytes long (about 1 BASIC line) but what a chore! After getting the memory bytes to show up on the screen and looping back again and again, the program needed something more to make it almost useful. Ah.... maybe more than the first page might be more exciting. So, all I needed to do was throw in a loop to scan "what-key-is-pressed"-type command (BASIC "GET" command) and screen out all keys except the cursor up/down key, increment the page number, then loop back and poke it to the screen. Sounds simple until you realize it meant moving all the coding down an unknown number of locations and correcting all references to specific addresses in the code, just to insert a few more bytes. THEN you realize, to your horror, you didn't move it down enough. (heavy sigh!!!.....Yea, yea, I know, a guy is SUPPOSED to plan out what he is doing first, then proof read and check his logic, then go program it in, but I like to do it my way!). Anyway, my hopes for ML dwindled and BASIC was an appreciated, SLOW, but appreciated comfort.

My desire to learn ML was suppressed until, browsing through Software Supermarket, I happened to spot a book called "128 Machine Language For Beginners" by Richard Mansfield, (Editorial Director-Compute Magazine and author). In it they include a program called "L.A.D.S." (no, not an dreaded Irish disease) that he said would make ML programing as easy to edit if not easier than BASIC (Ha ha, chuckle, chuckle). No sooner said than bought!

At home I read the book from cover to cover and spent hours typing in L.A.D.S. (using Compute Gazette's MLX, an ML entry program...See, I told you it's easier to read hieroglyphics). (If you get this book and want the program, please let me save you many hours of typing and ask me for a copy.)

To make a long story short, IT WORKS!...and BOY, DOES IT WORK! It not only makes it easy to insert coding, but lets you use many BASIC utilities such as MetaBasic, BASIC AID, renumbering, find and replace, merge, seek, scan, etc. It allows you to document your code (try that in assembly) so you can make sense out of it later. In short, it is an ML

programmer's delight! It allows you to relocate coding (not always an easy or pleasant task) by simply changing a number at the beginning and reassembling. While it assembles, it checks for typo errors (not logic errors) and reports any to you.

Before we get to far, I want to give credit for Richard's efforts in this book. He totally turned me around in my way of thinking of ML. He takes you through the fundamentals of addressing modes, computer math, instruction set and into ML equivalents for BASIC commands (a nice touch). He also gives a brief memory map of the 128. L.A.D.S. (Label Assembly Development System) is itself a machine language program that looks at your coding that resides in BASIC's program text space. This implies that you can list it as if it were a BASIC program. In fact, this is it's prime advantage, for it can thus use many BASIC utilities that programmers are familiar with.

However, you cannot simply load "L.A.D.S.source" and type run, or you will get a syntax error. The coding, like BASIC, has line numbers, BUT that is where the similarities end. The rest is similar to ML coding. For example-

```
10 *= $1300 ;or 4864 for decimal
15 SCREEN = $0400 ;or 1024 for decimal
20 .S ;view L.A.D.S. assembly on screen
30 .O ;assemble out to memory
40 LDX #0
50 LOOP LDA $0000,X:sta screen,x
60 INX:BNE LOOP
70 RTS
```

Please notice that there is no mention of memory addresses other than the first line. To relocate you just change the first line and reassemble! This program will assemble at \$1300 (4864 decimal) and proceed from there. You do not keep track of obscure hex numbers for your routines, just name them to your liking, then refer to them by name, as in line 50.

Also advantageous is the ability to assign memory locations to labels as in line 15. Any references to SCREEN automatically substitutes \$0400 on assembly. Another advantage is that you are not restricted to hex numbers only. If you like decimal, just insert it. If you like hex, proceed with the usual '\$' sign. Line 20 tells L.A.D.S. to let us see what it is doing while it assembles on the screen. Line 30 tells L.A.D.S. to actually put the finished coding into memory starting at \$1300. Line 40 is the beginning of your code. Any labels that you want defined to refer to memory have to be put in the first few lines. Otherwise, you are free to make up labels for your subroutines. A very useful feature is the '.byte' PSEUDO-OP (all commands to L.A.D.S. start with a "." to separate it from an ordinary label). This allows you to insert and label data tables, messages, info, work spaces, etc.

```
e.g. --80:HELLO .BYTE "Hello there"
      90 .BYTE 0
```

You can then make a loop to retrieve the individual letters, poke it to screen or use the KERNAL print sub \$FFD2 until you hit the 0 byte which is a signal to loop out with "BEQ". '.BYTE' allows you to insert alphanumeric and L.A.D.S. will convert them to ASCII (neat huh!). You COULD use it to insert decimal no.'s from 0 to 255 to represent the message you want directly into memory, but why go through the bother if L.A.D.S. does it for you.

All in all the L.A.D.S. assembler makes ML quite a bit more like BASIC, especially where editing is concerned. I would go so far as to say it's like another language. This is a very good way to learn ML. You have so much more flexibility compared to a monitor. It gives you more power to insert if you want to add more to your code, without having to go into your monitor and dutifully shift all your code down, making more errors (Murphy's law). It gets your coding tight and sequential without a bunch of NOP commands between routines.

There is no way I would try to develop ML routines longer than 30 bytes using a monitor. Use the monitor for debugging and small,....very small, programing purposes. Only a label-based assembler like L.A.D.S. allows me to concentrate on the program being developed with the comfort of known useful utilities like Commodore's screen editor and BASIC AID or METABASIC are at my side to boot.

NEW Disk Library Additions -

Scratch 'n' Save

E. Brown

COMP UTIL. 5 #CE

UGS 128 PGMS #12

UGS LOADER
UGS DATA

FUNCTION KEY
VISIBLE 64.C
TRANSFORMER
LINE COUNT
WINDOW WIZARD
FLNKSPD LIST CRE
SCREEN WINDOW
FREE RAM CHECK
BIG PRINT
BAR CHARTER
RAM DISK
BUG REP/64
INSTA BUG/64
FLANKSPEED/64
REVISED MLX
LITTLE WINDOWS
EXT BACKGROUND
SPLIT SCREEN
PROOFREADER 64
BUBBLE SORT
SHELL SORT
BINARY TABLE
BINARY QUIZ
SNOOPER
DATADJUSTER
FOOLPROOF INPUT
CROSS-REFERENCE

MEMORY CHART/64
GET SUBROUTINES
MONEYOUT
USING JOYSTICKS
HEX/DEC/BIN
LINE CROSS REF
AUTO LINE
BASIC MESSAGE
CATHODE RAY PRGS
VERTICAL MESSAGE
STRINGS.C2
RND GENERATOR.C2
SCROLL
PRINTDOC
SCROLL.DOC
ML MAILBAG
CRUNCH
DATASTUFFER
STUFFERBAS
TINY SORT
SORT TEST
TEXT FRAMER
GAMESPORT TEST
BUBBLE SORT 2
MOD BUBBLE SORT
SHELL SORT 2
IMPATIENT INPUT
ML BASE DEMOS
REMEDY
DATAMAKER
LINE EXTENDER
STP LOADER
STP SYS49152
STP SYS828
TICKERTAPE 64
AUTO-DEFAULT.BAS
SUPERNUMBER.BAS
SUPERNUM ARRAY

NEW DRAW N PAINT
BINARY MOON
S-P-C
JOG 128
KTOH LOGGER V1.2
DISK GOODY V5
HISPINS
MMF READER
3 COL DIR PRTR
DATA CREATOR
FAST SHUFFLE
SCROLLER 128
ALT-KEY
MULTI RAM
GALACTIC CAB CO.
STARTUP
PLANET DUEL
LIST FORMATTER
INFOFLOW
FRACTAL MAKER
EXT-BACK
SPEEDSCRIPT 128
SS FILE CONV.
NUMBER PLEASE
VIDEO SLIDE SHOW
APPENDER
80COL MAGIC DEMO
SCREEN DUMP 128
DUAL DUMP 128
WINDOW DUMP 128
SPRITE DUMP 128
SECTOR ED.80/128
GUMBALL RALLY
DATETIMESTAMP128
BARRICADE/128
PIZZA RAID/128
POSTERPRINTER128
DS 1571 RESTORE
POWER PAUSE/128

Take note! We have a new COMPUTER UTILITY disk this month (number five), a new 128 programs disk (number 12), and if I receive the programs early enough, a new GAZETTE disk (#25) which will have to be listed in next months MONITOR as this month's issue will have already gone to press.

Before I go on, I would like to mention the second-last program on this months listing of the 128 disk. If you're like me and switch back and forth constantly between the 64 and 128 mode, you may inadvertently validate a two-sided disk in the 64 mode. It appears, at this point, that all those programs on the second side are lost forever. Not so. The short (two block - one block is used to identify it) program will restore the pointers (or whatever) to re-establish all program listings as they appeared before the improper validation. Those people that purchase GAZETTE magazine will also find the 128 version of SPEEDSCRIPT on this disk. Besides the 80 column feature of this program, it also utilizes some of 128's special keys (like CAPS LOCK) etc. It would be wise to read the article before using it, to familiarize yourself with the changes. However, there are very few changes in the program and, if you have used the latest 64 version (3.2) of Speedscript, you could use most of this program's commands with the same keys.

How many of you will be asking SANTA for a new (another) disk drive for your 64. Besides those mentioned in last month's article, you have one more choice. It is the new 1541 II. It's identical in all physical and electronic ways and compatible with the 1541. Generally:

- it is smaller in size and its decor is 64C styled
- the power supply is encased seperately (less heat)
- all (?) previous bugs have been eliminated
- it is in stock in Regina (Bart's, for one)
- sorry, I forgot to ask the price (perhaps a little more than a 1581?)

A VERY MERRY CHRISTMAS and A HAPPY NEW YEAR

Next Meeting:

ALL ABOUT YOUR DISK DRIVE!

This major presentation will feature a comprehensive look at the 1541 drive from the "inside out"!

You'll hear: how it stores programs, DOS commands, how to tell when it's "ill", home maintenance, changing the device number, a quick "look" at the 1571 and 1581, and a peek at some software protection methods in use today.

Next C.U.G.S.
Meeting
Wednesday
January 6/88

SIR RICHARD'S BASIC!

This article is the second in a series examining the different file types and the programming involved with each type. In the first article, I outlined the different file types and generally how a disk is set up to handle these file types. In this article I am going to examine program files.

The program file is the most common file type. Program files store BASIC or assembler language programs. On a diskette directory, the characters PRG after the file name indicate a program file. If you use a sector editor to examine a disk containing a program file you will find some very interesting things about the storage of this file type. First of all, on track 18 (the directory track) you will find that reference to the file contains first of all a hex \$82 (decimal 130). This indicates a file is a program file type. If the file contains a hex \$C2, the file is locked (unscratchable - has a > sign before PRG in directory). Programs that lock or unlock files simply make this change. If you want to try this change for yourself using a sector editor, try it first on an unneeded disk as some sector editor programs make errors in saving sectors back to the disk.

Following the file type, there are two bytes which give the location of the first block of the program file. The first program on a disk is generally stored on track 17 sector 0. This would show up as hex \$11 00. Following these two bytes is the file name.

If you go to the first block of the file, you can trace the file through on disk. A program file has the first block containing a two byte link to the next block of program and then a two byte load location. A machine language program could contain any values here depending on where the program is placed in memory. For example, a hex value \$00 00 would indicate a machine language program whose first value is stored in decimal location 49152. Most BASIC programs would contain \$08 01 which is the start of BASIC on a C64. (Note: this value is different for different computers which have the start of BASIC in a different location - a program stored on disk from a PET would have \$04 01 as the load location - more about this later.) The following 252 bytes of this block contain the program. A machine language program would just consist of the bytes used to make up the program. A BASIC program consists of the lines of program stored as they would be in the computer. The first two characters are a pointer (low-byte/high byte order) to the location of the start of the next line IN THE COMPUTER'S MEMORY. Following this is the BASIC line number in low-byte/high byte order. The characters of the line then appear with BASIC keywords TOKENIZED. The end of each BASIC line is marked by a hex \$00. Successive blocks are identical to the first block except they contain a two-byte link to the next block and then 254 bytes of program (the load location is not required). The last block contains a two-byte count of the number of bytes used in this block. The end of a BASIC program is marked by \$00 00 00. The first \$00 is the marker of the end of the last line and the following 00 00 indicate the end of the program.

A BASIC or Machine Language program can be loaded using two very similar load commands. The first is: load"filename",8 This is a relocatable loader. It allows the C64 to be compatible with other Commodore computers. With this load, the program is loaded into the start of Basic location (\$0801 or 2049). The load location on the disk is ignored and for a BASIC program all line links are recalculated and changed as they are loaded to fit the C64's memory configuration. The second load command is the non-relocatable loader (LOAD"FILENAME",8,1). This causes the program to be loaded into the location specified on the disk file's load location. This allows machine language programs to be placed in memory where they belong. Many games for example need a load"filename",8,1 and then a sys49152. The load puts the game at location \$C000 and the sys is to access the machine language program at that location. If a BASIC program is loaded with '8,1' and has something other than \$08 01 in its load location in the disk file, some strange things can happen. A program saved with a PET for example will load the program into \$0401 which just happens to be the start of screen memory on the C64. As a result, the lines of program will appear on the screen and then run into the normal BASIC memory location. Such a program cannot be used and often results in the computer 'hanging up'. Some other computers will cause the program to be invisible - it is there but at a location other than the start of BASIC. On the CUGS disks, our loader program uses '8,1' for all loads. This meant we had to load many programs with just '8', scratch them off the disk and then resave them so that a '8,1' load would put them into the correct location.

If you have a PET computer and want to load a BASIC program made on a C64 into it, there is a way. Follow the sequence as follows:

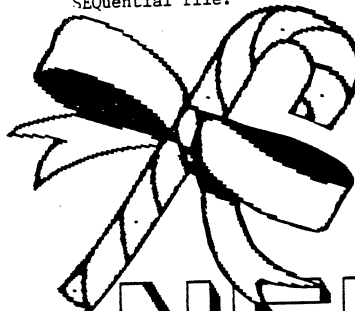
```
load the file normally
enter: 0 rom <press return>
enter: poke 1026,8 <press return>
enter: 0 <press return>
enter: list - the program should now appear.
```

This sequence creates a line 0 and changes the link so that it now points to the C64 program. Erasing line 0 causes the computer to pull everything down to replace the line and as a result the program now resides at \$0401 - the load location for the PET.

A separate tidbit - in a '8,1' load, the value 1 can be replaced with any other number and will work the same. LOAD"FILENAME",8,8 will work the same as LOAD"FILENAME",8,1.

Program files are saved using the save command. For a machine language program, this is often done through an assembler program or a monitor program. It may also be done through a BASIC loader but I will leave discussion of the process of doing this until after I have examined the statements needed which will be examined in sequential files. For BASIC programs, the save command consists of SAVE"FILENAME",8. When a BASIC program file is saved on disk, the load location is put onto the first sector so it can be loaded back to the same location later. When you save a BASIC program, it is always a useful exercise to verify that the save was completed correctly. You don't want to lose a lot of programming just because you incorrectly named the file to the same as an existing file name. Unfortunately, with the C64, you won't know something didn't save properly until you try and use it the next time. To verify a save, enter the following when you save the file: SAVE"FILENAME",8:VERIFY"*",8 <press RETURN>. This will save the file and then immediately compare the file just saved with what is in the computer's memory. You will either get 'OK' or 'VERIFY ERROR' which will tell you if the save was done properly.

In the next article, I will examine SEQuential files and some of the special things that can be done with this file type. I will also look at some different ways data can be saved using a SEQuential file.



NEW

FROM C.U.G.S

CUGS exec. member Steve Bagues has designed and drawn a blank overlay for the C64 and the 128 keyboard. The overlay is reproduceable on light bristol board or paper, and is a handy way to identify special key presses in common applications programs. Anyone interested in obtaining some of either overlay should contact an executive member before the first of January. They are available on plain paper (\$.50 each), on white 2-ply bristol board (\$1.00 each) or bristol board LAMINATED (\$1.50 each). Please make your request using the form below. They will be available at the January meeting C.O.D. (or other pick-up arrangements can be made).

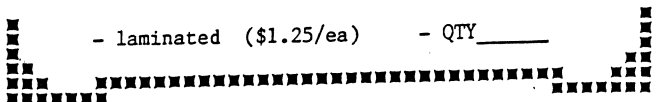
NAME: _____

TELEPHONE NO.: _____

- on plain paper (@\$.25/ea) - QTY _____

- bristol board (@\$.75/ea) - QTY _____

- laminated (\$1.25/ea) - QTY _____



Computerfest '88

COMPUTERFEST - FEB. 7, 1988
(sponsored by - Apple II user group)

Vagabond Motor Inn (Spanish Ballroom)

Noon - 5:00 pm

Displays by major retailers of computer hardware and software.

User groups will each have their own display.

A "flea market" of computer items.

Come and visit our CUGS booth at Computerfest.

Watch for more details in the January issue of 'The Monitor'.



Meeting Dates to June, 1988.

Wed. Jan. 6
Wed. Feb. 3
Wed. Mar. 2
Wed. Apr. 6
Wed. May 2
Wed. June 1

Mark these on your calendar!

Meetings start at 7:00 pm

Prima Donna:

My Ami, the Vice!



To Buy or Not to Byte

That is a question I have asked myself several times in the last few weeks. Why?... because I found myself looking at an AMIGA 500 at Software Supermarket one day last month and found myself wanting the machine. Why?... because Bart was selling off 500's for \$1595.95 (approx.) - including the 500 the drive, monitor, mouse, and printer. This sounds like quite a deal and it is a deal. However....

Having a new machine like the 500 is like the C64, 6 years ago. There is not much in the way of "GOOD CHEAP SOFTWARE". Yes, there are some good programs out there but not the quantity like the 64/128 has. At this point in it's history, really good software is at quite a high price. Usually in the range of \$100.00 and up.

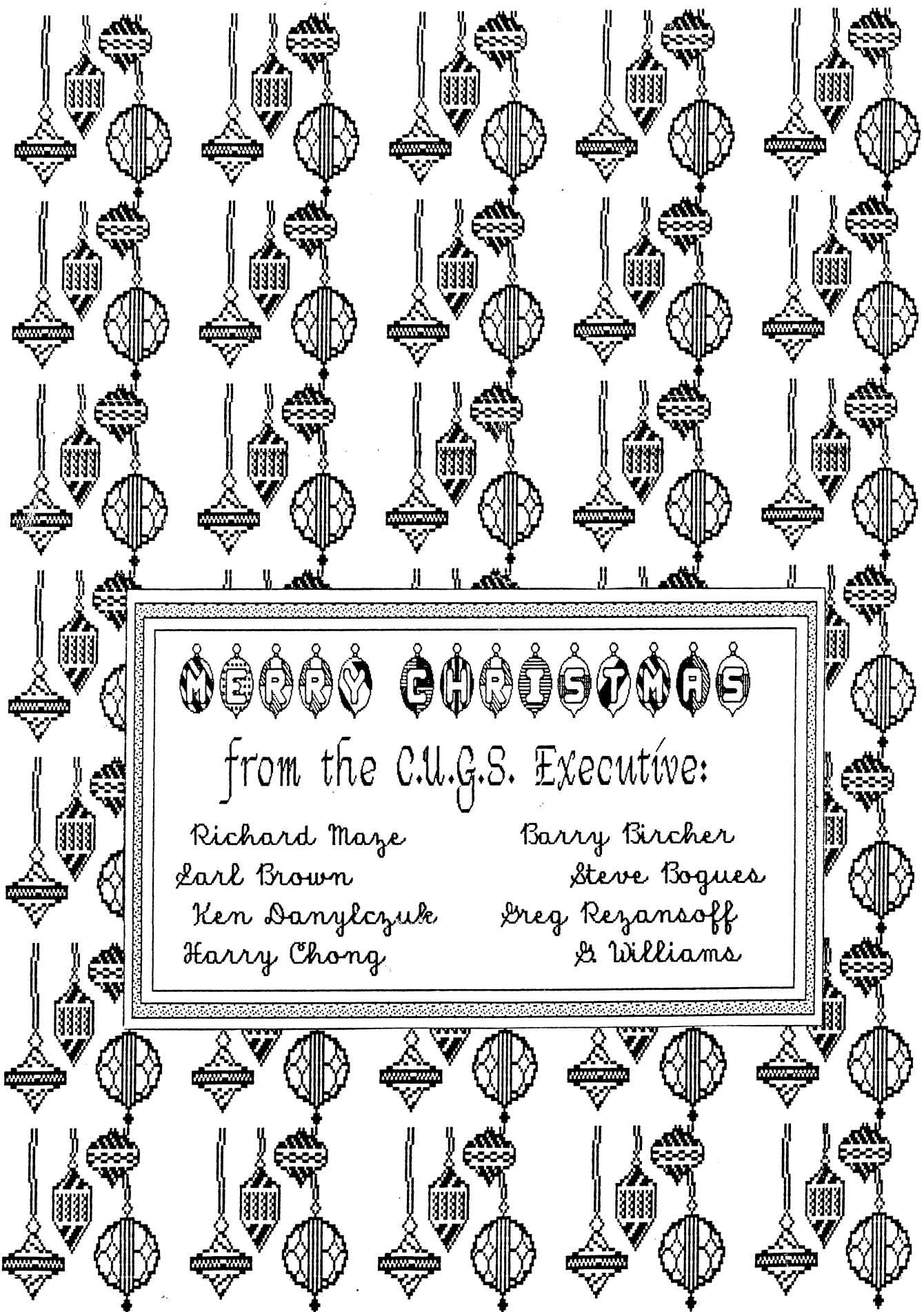
Don't get me wrong, the Amiga line has really impressed the bits out of me, as well as the Atari (Heaven forbid!). The point is, after several years behind the 64 and then the 128, I feel very comfortable behind the keyboard and the memory arrangement. Introducing the newer 68000 based computers into my life right now is not going to improve my primary use of the things, mainly word processing and budgeting. Yes, you can do both as well as several other programs AT THE SAME TIME, but really, in the home market I see limited use for that feature, however impressive it is!

I recently picked up an AURA newsletter and found it interesting and informative. Like all other computers on the market, it has it's share of bugs, being a new computer. Supposedly, this newsletter was produced by the editor's Amiga and I cannot really see the difference it made to their newsletter. Our MONITOR seems by far better assembled (Thanks to Ken Danylczuk). I can't get over the sharp graphics compared to the 64/128 (magazines make a point of putting 64/128 and Amiga graphics side by side). Again however in the 2 1/2 years in programming, I have not yet begun to look at the sprites and bitmap graphics that are in my computer, except when I throw in the odd game to play.

All in all, and after several hours of talking myself out of it, I realized that the introduction of the Amiga 500 was like another fancy car being introduced on the market to fill a gap. Some people like Chevy's, some Dodge Ram's, some like Cadillac's and some like Rolls Royce. There is something about the personality of cars and computers that are a lot alike, that attract different people to different makes and models. They all get you from point "A" to point "B", some faster than others and some in high class, some in economy class.

For my money, I think I will keep up to my trusty 8 bit 64/128 and get the job done with the equipment I have now and let somebody else take their lumps with the Amiga (until all the bugs are out, if that's possible, or when Commodore comes out with a newer 32/64 bit superconductor parrallel processor) I may then retire my 128 and splurge on it.





M E R R Y C H R I S T M A S

from the C.U.G.S. Executive:

Richard Maze

Earl Brown

Ken Danylczyk

Harry Chong

Barry Bircher

Steve Bagues

Greg Rezanoff

& Williams