EDITORIAL:

Since I joined the C.U.G.S. executive 2 years ago, I've been impressed by the general concern on the part of all executive members to keep our meetings and our club's offerings up-to-date, practical, interesting and novel. A look back at our meeting agendas will show that clearly. And we ain't done yet. Next month we will be trying something new (again!) for our club members (at least in the past few years).

In the early days of computers (especially COMMODORE computers), software was sparse, even non-existent. Those dedicated few who had machines naturally met together to share their discoveries about their machines (in the absence of decent operating instructions) AND to share software (always in short supply). Thus were planted two seeds - the roots of the COMPUTER USERS' GROUP - a group of users gathering together for growth in understanding (just like us!); and the first attempts at a SWAP SESSION (exchanging user/public domain software for the benefit of all concerned).

In the past couple of years commercial software has increased in quantity (and quality) to the point where SWAPPING is no longer NECESSARY to obtain a variety of useful programs. This proliferation of commercial material has had a two-fold negative effect on what should be a time-honoured tradition of us computer lovers. First, the vast expanse of brightly-packaged, advertised, and reasonably-priced software has hidden (NOT eliminated) the work of the PUBLIC DOMAIN programmer from the general computer user. The programs are still being written - they're still being shared - but not as evidently as 6 years ago.

Also, because computers became so widespread, a different kind of computer devotee appeared - the "cracker" (NOT "hacker" as they'd like to be called) - a person dedicated to pirating software and "sharing" it with friends. "Sharing" software took on sinister overtones of devious, arrogant computer wizard dominating the latest commercial game! This negative aspect also did much to "hide" the legitimate exchange of decent, public domain material.

But the exchange of good, "home-made" computer software IS seeing a revival! PUBLIC DOMAIN software is original-work, a prvilege which has been given for distribution to anyone interested. FREeware and SHAREware are variations on PUBLIC DOMAIN software; they are programs offered to the public freely, with the request that anyone trying the work and finding it useful send some amount (usually nominally small) to the author. Most public domain/shareware material carries the restriction that the work MUST NOT be sold, or used as part of works which will be sold commercially.

Anyway, I seem to be rambling. Next month's meeting of C.U.G.S. should be fascinating for anyone interested in extending their software library. C.U.G.S. will operate their first (annual?) SWAP NIGHT for members only! Admission will be a membership card! We'd like to encourage everyone to bring along some blank disks AND some PD software to swap. (No disks - no swaps - short evening!) Please try to avoid bringing material currently in our library (no fair!) and be prepared to "share" it with our librarian, Earl. If it's checked and sorted it'll be added to our regular disk library! This evening could be the greatest thing since C.U.G.S., or a night we'd rather not remember - it all depends on you! (Gee, sounds like a great line for a song title!) See you next month - signed Trader Dan!
The MAY meeting of CUGS will be held WED. MAY 4.

This meeting will be set up as an opportunity for CUGS members to share public domain 64/128 software.

Each member is requested to bring a disk(s) containing your favorite public domain program(s) and a supply of blank disks (limited supply for sale at meeting @ $1.00 each).

Computers and dual drives will be made available for copying disks. A copy of your program(s) will be made for inclusion in the CUGS library.

To speed up the copying process, we ask that all disks be in 1541 or single-sided 1571 format.

Club members may bring commercial software for sale or trade, as long as the software is original software only and they assume full responsibility for the software (we won't have a supervised sale table or area).

Since this activity is for CUGS members only, you may be requested to show your membership card. (Memberships will be available at the meeting).

Coffee, cookies, conversation and advice will be provided.

The VIC Chip sees graphics in 4 parts and reconstructs a picture on the screen. The first (main) part involves 8000 bytes called the BITMAP. The chip looks at the current character position within the BITMAP, and extracts the byte it wants to scan on screen. It breaks the byte into 4 2-bit NIBBLES. Using the 2 bits (corresponding to 2 pixels on the screen), it searches the colour it is to display using the codes held in the nibbles. Since 2 bits are able to count to 4, we are able to display 4 colors within a character position. The nibble code simply tells the chip where it is to get the colour source code corresponding to that character position. We therefore need to store the colour codes to be displayed in another part of memory (within the chip's addressing range). The chip has only 14 lines to address with, so it can only access 16K at a time. What this means is that our info needs to be within that range.

The background byte we are most familiar with ($3281/$$020) we will call Color #00. It is common to all screen positions and must be one of the 4 colours to be displayed. Since color ram (1000 bytes from 53296/$$800 to 56295/$$DBE7) is not movable, we can count on this being here all the time. We will call this Color #11. This accounts for 2 colors... so where do the other 2 come from?

There are 1000 bytes (usually below the bitmap ram) that are encoded to hold 2 color codes within one byte (since 4 bits can count to 16 we only need 4 bits per color). The upper 4 bits hold the code for what we will call Color #01, and the lower 4 bits Color #10.

So, there we have it - Color 00 01 10 11 - 4 colors. Now, back to the bitmap. This is the heart of the multicolor mode. The VIC chip looks at the bitmap and determines the 2 pixel group it is to display. It decodes the 2 bits and displays the color pointed to by the nibbles.

00 = display color in Background register. ($3281/$0020) 1 byte
01 = display color in upper 4 bits of color map ($1024/$0400) 1000 bytes
10 = display color in lower four bits ($1024/$0400) 1 byte
11 = display color in color ram ($5296/$DB00) 1000 bytes

The bitmap as well as the color bitmap is able to be moved around to one of 4 video banks seen by the VIC chip. The background and color ram cannot be moved, so they stay put. A KOALA picture file consists of a first byte of unknown use (if you know what the first byte does, please let me know!); a second byte to hold the border color, then the 8000 byte Bitmap, 1000 byte color bitmap, 1000 byte color ram, then the background color byte (10,003 bytes total).

KOALA READER (BASIC 128)

100 BANK15
110 CATALOG"PIC4"  :REM SELECT AND OPEN FILE
120 PRINT"CURSOR TO FILE & HIT RETURN"
130 INPUTS:AS=MIDS($A,7,15)
140 OPEN2,8,2,AS
150 GET#2,AS,POKE 53280,ASC(AS):REM LOAD BORDER COLOR
160 GRAPHICS3,1
170 FOR J=DEC(2000) TO DEC(3333):REM LOAD 8000 BYTE BITMAP
180 FOR K=DEC(1000) TO DEC(1107):REM LOAD 1X COL. BITMAP
190 GET#2,POKE1,ASC(AS):NEXT
200 POKE1,216,255
210 POKE,PEEK(1)AND 254 :REM COLOR BANK (2 AVAILABLE)
220 FOR J=DEC(3334) TO DEC(5665):REM POKE TO COLOR MEMORY
230 GET#2,AS,POKE1,ASC(AS)AND 15 :REM LOAD BACKGROUND COLOR
240 NEXT:POKE1,PEEK(1)OR POKE 216,160
250 CLOSE2

KOALA PICTURE DISPLAY (BASIC+ML 128)

10 BANK15:GRAPHICS3,1:GRAPHIC0 :REM CLR SC & MOVE BASIC
20 BLOAD"KOALA-R-ML",10,D,4684
30 PRINT"PLEASE INSERT KOALA DISK":GETKEY AS
40 CATALOG"PIC4"
50 PRINT"CURSOR UP TO FILE & HIT RETURN"
60 INPUTAS
70 AS=MIDS($A,7,15)
80 OPEN2,8,2,AS
90 GET#2,AS
100 GET#2,AS,POKE 53280,ASC(AS) :REM BORDER COLOR
110 GRAPHICS3,1:SYS 4684
120 GET#2,AS,POKE 53281,ASC(AS) :REM M.L. TO DUMP FILE
130 CLOSE2 :REM BACKGROUND COLOR
140 GETKEYAS:GRAPHIC CLEAR
150 POKE 53296,13:POKE 53290,11:POKE 241,13
160 PRINT"RETURN TO DEF. COLORS"
170 PRINT"DISPLAY ANOTHER? Y/"N"
180 AS="Y":INPUT AS
190 IF AS=5 THEN CLR:END
190 GOTO 40
200 END
KOALA M.L. L.A.D.S. SOURCE 64

KOALA PICTURE DISPLAY (BASIC

10 REM 10 = $1300
20 .S 20 REM
30 .D KOALA-READ64-ML KOALA-R64-ML
40 LDX #2 ;JSR $FFC6
50 LDX #$FF ;STX #216
60 LDA #01 ;AND #$FF ;STA #01
70 LDA #$20 ;STA STOREIT+2
80 LDA #$00 ;STA STOREIT+1
90 BITMAP JSR $F8E4
100 JSR STOREIT
110 LDA STOREIT+2:CMP #$3F ;BNE DONE1
120 LDA STOREIT+1:CMP #$40 ;BNE DONE1
130 JMP COLORMAP
140 DONE1 JMP BITMAP
150 COLORMAP LDA #$1C ;STA STOREIT+2
160 LDA #$00 ;STA STOREIT+1
170 COLORBIT JSR $F8E4
180 JSR STOREIT
190 LDA STOREIT+2:CMP #$1F ;BNE DONE2
200 LDA STOREIT+1:CMP #$5E ;BNE DONE2
210 JMP COLORRAMO
220 DONE2 JMP COLORBIT
230 COLORRAMO LDA #$0B ;STA STOREIT+2
240 LDA #$00 ;STA STOREIT+1
250 LDA #$FF ;STA $DB
260 LDA $01 ;AND #$FE ;STA $01
270 COLORRAM JSR $F8E4
280 JSR STOREIT
290 LDA STOREIT+2:CMP #$DB ;BNE DONE3
300 LDA STOREIT+1:CMP #$EB ;BNE DONE3
310 JMP SET216
320 DONE3 JMP COLORRAM
330 SET216 LDA #160 ;STA $DB
340 LDA $01 ;ORA #001 ;STA $01
350 JSR $FFCC ;RTS
360 STOREIT STA $2000
370 INC STOREIT+1 ;BNE DONE
380 INC STOREIT+2
390 DONE RTS
400 .END KOALA-READ64-ML

KOALA M.L. L.A.D.S. SOURCE 64

prizesprizesprizes

At each CUGS Meeting during 1988, there will be a computer-generated draw for a winner of a prize.

RULES:

Paid-up members for 1988 ONLY are eligible.

Draw is made at the end of each meeting.

The winner must be present to claim the prize. If a member NOT present is drawn, the draw will be made again until a winner is found.

Prizes are to be accepted "As is" no substitutions permitted!

The membership list will be updated at the break and new members will be eligible.

Prize for April: THE HOME MANAGER (program)

March winner: BARRY BIRCHER

prizesprizesprizes
Sir Richard’s BASIC:

by Richard Maze

In last month’s article I examined the procedure involved in creating a relative file. The follow-up to that article is a program that actually uses the relative file that was created. The program that follows is a simple, no-frills relative file handler. It assumes that the file ‘ADDRESS’ has already been created on disk. This program, along with the one from last month, would constitute the start of a file-handling system. All that needs to be added is more flexibility in creating the file and the ability to report to paper.

100 REM USE RELATIVE FILE ** ADDRESS **
120 REM 200 RECORDS
140 REM RECORD LENGTH = 83 CHARACTERS
170 REM SET VARIABLES
180 : POKE 53272, 23:REM LOWER CASE
190 : POKE 53280, 15:POKE 53281, 15
200 : DNS$="HOME CRSR DOWN 23 TIMES"
210 : PDS$="$<24 SPACES"
220 : BLS$="$<25 SPACES"
230 : DIM LAS(6), DAS$(6): CR$=CHR$(13)
270 REM TITLE
290 : GOSUB 2240
310 REM >>> MAIN LOGIC <<<
330 REM OPEN FILE
350 : OPEN 15.8.15:REM COMMAND CHANNEL FIRST
360 : OPEN 2,8,2:"0:ADDRESS"
380 : REM GENERAL SET-UP
400 : GOSUB 620
420 : REM MENU
440 : PRINT "CLR";LEFT$(DNS$, 9);TAB(6)
450 : PRINT "BLACK WHAT OPTION? REDSELECT NUMBER"
460 : PRINT TAB(10);"DOWN RED 1 BLUE SEE /
CHANGE A RECORD"
470 : PRINT TAB(10);"DOWN RED 2 BLUE ADD A RECORD"
480 : GET G$:IF G$="E"THEN GOTO 240
490 : G=VAL(G$):IF G>1 AND G<6 GOTO 480
500 : ON G GOSUB 840, 1400, 230
510 : GOTO 440
530 : REM QUIT
550 : CLOSE 2;CLOSE 15
560 : PRINT "CLR"
580 END
590 : REM GENERAL SET-UP
620 : REM GET FIELD LABELS
640 : FOR I = 0 TO 6
650 : READ LAS$: LAS$(I)=LEFT$(LAS$+BLS$, 13)
660 : NEXT I
680 : REM FIND EMPTY RECORD
700 : RN = 1
710 : GOSUB 1970:REM POSITION TO RECORD
720 : INPUT#,$
730 : IF LEFT$(LAS$(1),1)="E" THEN ER=RN:GOTO 780
740 : GOSUB 1970:REM POSITION TO RECORD
750 : RN = RN + 1:IF RN < 201 GOTO 710
760 : ER = RN
780 RETURN
790 : REM SEE/CHANGE A RECORD
820 : REM GET A RECORD NUMBER
840 : PRINT "RED WHAT RECORD DO YOU WANT?"
850 : PRINTTAB(8)"BLUE DOWN ENTER THE NUMBER (1-200)"
860 : PRINT TAB(8)"DOWN AND PRESS RETURN"
870 : INPUT " ":LEFT LEFT LEFT$: RN
880 : RN=VAL(RN$):IF RN<1 OR RN>200 GOTO 840
900 : GOSUB 1720:REM GET DATA
920 : REM DISPLAY DATA ON THE SCREEN
940 : PRINT "CLR":GOSUB 1820:REM LABELS
950 : PRINT "HOME RED RVSON RECORD NUMBER":RN:"LEFT "
CAPUTE'S G.D.G. OPCODES

by Barry Bircher

Here is one of CAPUTE'S most popular and useful utilities ever published in a magazine since it was first published last week. The program works itself in the most useless place you can think of in the computer so you don't have to worry about crashing it (All it takes to crash it is to be near an electrical outlet). The program can be used in both the monitor and in BASIC as it is wedged in a main KERNAL vector "CRaSH", and therefore is not affected by some idiot hitting RUN/STOP=RESTORE.

Here are some useful opcodes to use when you boot your computer with my new GOSH DARN GOOD MACHINE LANGUAGE, language extension. It can be used with all existing CPU's on the market today. It will auto-configure itself to your system, so there's no need to reloacte. Since GOSH DARN GOOD is totally M.L., you will have to use MLUP (elsewhere in this issue) to enter it. When you have MLUP up and running, it will ask you for a staring address. You should answer 2300 and end it at 2320. To introduce you to my GOSH DARN GOOD operating system (GEDS compatible) here are some useful opcodes to use while in monitor mode. Program listing is on page 9. File at eleven.

ADC = Add with Carry
AND = Logical operator, using for masking out bits.
ASL = Automatic Shift To your Left
BCC = But Can't I Come
BCS = But Can I Say
BEO = Before Ernie Quits
BIT = Back In two jiffies (in the biffig)
BMI = Beam Me In
BMU = Beam Me Up (Scottie!)
BNE = Buy Non-Expensive equipment
BPL = Branch if Processor Locks (very useful)
BRE = coffee BreaK
BRT = Logical operator eg. BUT IF it doesn't then
WHYN0T (see also NOT, WHY and HOWcome)
BVC = Branch if over-Crowded
BVS = Branch if overStocked (bypasses commercials)
CLC = Clear the Carry
CLD = Clear it Dammit!
CLI = Clear it, you Idiot!
CLV = Collect Leftover Variables
CMP = Call Mom & Pop
CPX = Collect Pension Cheque
CPI = Copy
DEC = To hit someone
DEX = To hit someone (plural)
DEY = Demand and Yell
EOR = Winnie the Pooh's donkey (kick start)
IF = Logical operator eg. IF this AND that BUT NOT
this OR this then WHYN0T (see MAY
IF, AND. OR, BUTTES)
INC = Stuff they use in pens
INX = What the washer does to pens and your clothes
INY = I know Not wht (logical operator
used for no knowledge)
JMP = What you do if this works
JSR = Junk Status Register
LDA = Long Day
LDX = Load and Cross
LDY = Load and Yell
LSR = Last Stupid Remark
MAY = Logical operator eg. MAYbe this, OR that
AND then again MAYbe NOT BUT then MAYbe so
(see also NOT, IF, AND, OR, BUT)
NOT = Logical operator eg. NOT this AND that
AND BUT NOT that (see AND)
NOP = NO Party tonight
ORA = Logical operator eg. Could be this ORA,
this OR maybe NOT this and NOT that
(see also BUT MAY NOT MAYbe)
PHA = Push Hard
PHP = Party Hardy Partner
ROL = short for Rolls Royce
ROR = Lion Sound (used by SID's)
WHY = Logical operator eg. if WHY then HOW did
it NOT work (see also HOW, NOT)

Commands Used by GOSH DARN GOOD in BASIC
B. Bircher

LPRINT = print on line L
HPRT = print Hard copy
LPRT = Letter Quality print on dot matrix printers
COLORS = set up 255 colors from a palette of 65535
CCOLOR = Select colors from color (see COLORS)
MSIZE = check on memory size in K's & add if necessary
SDOS = speed up disk access to 1 Giga byte/second
RAMDIS = Turn disk storage into contiguous RAM (see
SDOS)
TERM = Built in Terminal program with 640K buffer
ETREM = Speed up 300 baud modems to emulate 2400 baud
GCP/M = switch to CP/M plus
GOMS = switch to MS-DOS
GOBED = Sets alarm, shuts off computer after 11:00 PM
G0640 = adds 640K bytes to BASIC storage
G0128 = Switch to 128 mode
FFAST = switch to 4 MHz clock
SFFAST = switch to 20 MHz clock
SP = Switch disk to regular speed
LP = (Long Play) slows disk to get more
info on it. (see SLP)
SCL = (Super Long Play) gets up to 3.25 Gigabytes
per disk. (see also SP)
AMIG = Enable multitask mode
RECHRG = Recharge RAM backup batteries right now
UNCHRG = Erase your last 5 VISA transactions
JUST FOR THE RECORD

(OR IS THAT A FIELD?)

by Richard Maze

There are many terms that are associated with handling large quantities of data. Often these terms are used incorrectly resulting in confusion and a misunderstanding as to what a particular program will do.

The major terms used are: DATABASE, FILE, RECORD, FIELD, and CHARACTER. These terms represent a hierarchy of data organization. It is often easiest to look at them starting with the lowest level and working up.

CHARACTER: This is one letter or number entered as a part of data. It could be a letter of a person's name or one digit in a sales amount.

FIELD: A group of characters together which make up ONE DATA ENTRY entry. A person's last name or a sales amount might be examples of different fields.

RECORD: A group of fields make up a RECORD. The name, address, city, prov., and postal code for a person could be ONE RECORD.

FILE: A number of records is grouped together to form a file.

DATABASE: The grouping together of related files.

An example of this organization could be constructed as follows: If a company has a set of index cards for its customers, then, the entire set of cards would be a FILE. Each card, which contains information on one customer, would be a RECORD. Each space to be filled in on each card, (name, address etc.) would be a FIELD. Each field would be filled in using CHARACTERS. Now if the company had another parallel set of cards containing other information about its customers, (amount ordered, amount owing, etc.), this parallel set would be another FILE. The TWO SETS together would make a DATABASE.

One common misuse of terms is that programs that are really just file handling programs are called databases. For example, "a database to help you keep track of magazine articles" is really a file handler program - NOT A DATABASE. This is also true of commercial programs - many "database managers" are really 'file managers'. A true database manager program should allow data to be accessed from two or more files.

Another area that often creates confusion relates to sizes - FILE size, RECORD size, FIELD size. Strictly speaking, FILE size should be the NUMBER OF RECORDS that can be stored; RECORD SIZE is the SUM OF ALL the CHARACTERS that can be entered in all the fields; and FIELD size is the NUMBER OF CHARACTERS that can be entered in one particular field. If a program stores the data in memory, file size is limited by computer memory. Otherwise, unless the program limits it, FILE size should be limited only by disk space available. RECORD size is often limited to 254 characters (a convenient storage and transfer unit - 1 disk sector) but programming techniques can get around this. FIELD size is up to the user to determine but may be limited to 80 characters (input limits this) or 40 characters (screen width). The most important factor is that the sum of all field sizes cannot exceed the record size limit.

If you looking for a database/file program, it might be worthwhile to do some calculations based on the specified sizes. Often the number of possible records that is stated is only possible if there are minimum fields of minimum length. In most cases: larger field sizes and greater numbers of fields means less records that can be stored per file. You can't just calculate the available space on a disk to determine how much can be stored (664 blocks * 254 characters/block = 166656 characters) because many programs create auxiliary files to support the file which also take up storage space.

FOR SALE:

$70.00 O.B.O.

call BARRY 359-1925

ON-LINE WITH

Greg

BBS list as of March 11, 1988

$ = Pay BBS * = Temporarily Down

by Greg Rezansoff

(Ed. note)

The "Set" column below provides information on MODE (Full or Half Duplex), # OF BITS (7 or 8), PARITY (Odd, Even or None), and # OF STOP BITS (usually 1). The "Baud" column indicates the relative speed at which files can be transferred to and from the bulletin board. Most of this information is necessary to properly prepare your system to "log on" to the system desired.

<table>
<thead>
<tr>
<th>System Name</th>
<th>Phone Num</th>
<th>Baud-Set</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abyss</td>
<td>586-0721</td>
<td>300 FBN1</td>
<td>Dark Angel's FFBS</td>
</tr>
<tr>
<td>Bermuda Square</td>
<td>545-7678</td>
<td>2400 FBN1</td>
<td>RBBs</td>
</tr>
<tr>
<td>Bit Bucket</td>
<td>352-3226</td>
<td>2400 FBN1</td>
<td>4 Amiga,Atari,64</td>
</tr>
<tr>
<td>Dialogue</td>
<td>775-3587</td>
<td>2400 FBN1</td>
<td>Atari BBS</td>
</tr>
<tr>
<td>Digital Dimensions</td>
<td>586-6493</td>
<td>1200 FBN1</td>
<td>Atari 8 bit</td>
</tr>
<tr>
<td>DoubleCheck</td>
<td>525-0807</td>
<td>2400 FBN1</td>
<td>V. good Amiga brd</td>
</tr>
<tr>
<td>EMIS</td>
<td>586-5585</td>
<td>1200 FTE1</td>
<td>Unix multiplier $</td>
</tr>
<tr>
<td>EMIS 2</td>
<td>586-2035</td>
<td>1200 FTE1</td>
<td>Unix multiplier $</td>
</tr>
<tr>
<td>Fernando's</td>
<td>585-0298</td>
<td>9600 FBN1</td>
<td>Pyroto Mtc/PC *</td>
</tr>
<tr>
<td>Infobase</td>
<td>789-5463</td>
<td>1200 FBN1</td>
<td>PC clones only</td>
</tr>
<tr>
<td>Late Night Alt.</td>
<td>586-3285</td>
<td>1200 FBN1</td>
<td>Good convos/PC/Amiga/edl</td>
</tr>
<tr>
<td>Magic Fountain</td>
<td>586-2692</td>
<td>1200 FBN1</td>
<td>Small but growing</td>
</tr>
<tr>
<td>Micro City I</td>
<td>584-0747</td>
<td>1200 FBN1</td>
<td>Multiuser system $</td>
</tr>
<tr>
<td>Micro City II</td>
<td>584-0748</td>
<td>1200 FBN1</td>
<td>Multiuser system $</td>
</tr>
<tr>
<td>Negative Zone</td>
<td>565-8538</td>
<td>2400 FBN1</td>
<td>Good echos/convos Onyx</td>
</tr>
<tr>
<td>Onyx</td>
<td>347-1015</td>
<td>2400 FBN1</td>
<td>Great CP/M good PC support</td>
</tr>
<tr>
<td>Pirate's Galleon</td>
<td>775-3358</td>
<td>2400 FBN1</td>
<td>Growing PC system</td>
</tr>
<tr>
<td>Polestar Opus</td>
<td>586-1551</td>
<td>2400 FBN1</td>
<td>PC support</td>
</tr>
<tr>
<td>Probable Fate</td>
<td>525-1054</td>
<td>300 FBN1</td>
<td>Convers. filestfiles</td>
</tr>
<tr>
<td>Regina FIDO</td>
<td>347-4493</td>
<td>2400 FBN1</td>
<td>Great. PC/GP/M</td>
</tr>
<tr>
<td>R.A.D. BBS</td>
<td>945-7035</td>
<td>300 FBN1</td>
<td>Atari only BBS(NES)</td>
</tr>
<tr>
<td>R.A.D. II</td>
<td>949-6105</td>
<td>1200 FBN1</td>
<td>Good Atari 8 bit</td>
</tr>
<tr>
<td>Saskatchewan R O S</td>
<td>789-0690</td>
<td>1200 FBN1</td>
<td>Good Amiga/PC</td>
</tr>
<tr>
<td>Shadowland I</td>
<td>786-7883</td>
<td>2400 FBN1</td>
<td>Home of The Realm</td>
</tr>
<tr>
<td>Shadowland II</td>
<td>789-8989</td>
<td>2400 FBN1</td>
<td>Good Mac pic. area</td>
</tr>
<tr>
<td>State of Mind</td>
<td>352-1455</td>
<td>2400 FBN1</td>
<td>PC based RBBS(NES)</td>
</tr>
<tr>
<td>Switch</td>
<td>569-8888</td>
<td>1200 FBN1</td>
<td>Atari ST based</td>
</tr>
<tr>
<td>Tee Wun Kay</td>
<td>779-1237</td>
<td>2400 FBN1</td>
<td>PC support</td>
</tr>
<tr>
<td>Tesseract</td>
<td>757-5699</td>
<td>1200 FBN1</td>
<td>Excellent Amiga</td>
</tr>
<tr>
<td>Triple Diamond</td>
<td>775-3314</td>
<td>1200 FBN1</td>
<td>C-64 based AEBBS</td>
</tr>
<tr>
<td>Turbo</td>
<td>586-7560</td>
<td>1200 FBN1</td>
<td>Good convos 64&amp;PCP files</td>
</tr>
<tr>
<td>UltraFast Systems</td>
<td>545-5717</td>
<td>2400 FBN1</td>
<td>Amiga based (NEW)</td>
</tr>
<tr>
<td>Wizard's Wand</td>
<td>949-0178</td>
<td>2400 FBN1</td>
<td>PC supp./cheers</td>
</tr>
<tr>
<td>Datapac 300</td>
<td>565-0111</td>
<td>300 HN1</td>
<td>Datapac 300</td>
</tr>
<tr>
<td>Datapac 1200</td>
<td>565-0181</td>
<td>1200 HN1</td>
<td>Datapac 1200</td>
</tr>
<tr>
<td>Datapac 2400</td>
<td>565-6000</td>
<td>2400 HN1</td>
<td>Datapac 2400</td>
</tr>
<tr>
<td>U. of Regina</td>
<td>584-8085</td>
<td>1200 FBN1</td>
<td>U of R Develswitch</td>
</tr>
</tbody>
</table>
NEW C.U.G.S. DISK LIBRARY ADDITIONS:

by Earl Brown

For those of you that arrived at a negative number in line 20 of SCHEDULE 10 CHILD TAX CREDIT in the 1987 Income Tax program, the following change in line 403 of the SPECIAL, and line 767 of the GENERAL should be made:

403
or

Quite often, when writing a program, I type a new number over an existing line number and make the appropriate change to that line. For some reason, I forgot to make all the changes and the 'if' statement contained the wrong variable. Thank you, Richard, for bringing this to my attention.

For those members interested in the database program "MICROPHILE" which Richard is demonstrating this evening, you will find it on CUGS BUSINESS 9 DISK #81. Also included on this disk is another database program entitled "MINI-FILER" from Gazette Magazine (Feb/86) and a spreadsheet program "CALCAID" from Run Magazine (Nov/86) for those of you who purchase these magazines.

There are five additional disks for the 64 in this month's listings as well as three more for the 128. Also included in the listings is the latest Gazette disk which includes the remaining programs from January, all of February and March, and most of April 1986. Turbo Speedscript program that is included on this disk for the 64 won't work with my 1571 drive but does work with all 1541 drives. The boot and additional files allows this word processing program to fast-load and fast-save all your processed files.

Finally, one thing I forget to mention each month. The library has in its possession two books which members may borrow (in one-month stretches) from the library. They are:

1. THE COMAL HANDBOOK - a 310 page handbook on how to use the COMAL language to your best advantage. We have COMAL program disks in our library.

2. THE BEST OF TORPET MORE FOR THE COMMODORE 64 AND THE VIC-20 - This book covers many aspects of the 64 and Vic-20 and includes a disk with many useful programs on it.

I picked up a good looking printercover at February's COMPUTERFEST. The problem is it is too small for my FX-80 EPSON PRINTER. The dimensions of the cover are 14 1/2 inches wide by 12 inches front to back. Five dollars is what I paid for this cover manufactured by AMERICAN COVERS INC and if you can use it, it's yours for just THREE DOLLARS! OKAY!
PRESENTING:
the
Great Software SWAP

PLACE: NORTHWEST LEISURE CENTRE
DATE: WEDNESDAY, MAY 4, 1988 7pm

Open to all Commodore User Group members
(Please have your membership card).

Admission will be to card-holding members only. Members should come with PUBLIC DOMAIN or SHAREWARE material to swap.

NO COPYING OF COMMERCIAL SOFTWARE WILL BE PERMITTED.

Members wanting to offer legitimate used software for sale may do so on their own responsibility. We hope to see YOU there!