the U.G.S monitor

June, 1987

OBLIGATORY STUFF

PRESIDENT	Richard Maze
VICE-PRESIDENT	Ed Dietrich
SECRETARY-TREASURER	Gordon Glew
LIBRARIAN	Earl Brown
ASS'T LIBRARIAN	Randy Sloboda
EDITOR	Ken Danylczuk
ASS'T EDITOR	Greg Rezansoff
MEMBERS AT LARGE	Steve Bogues
	Harry Chong

THE MONITOR is published monthly by the COMMODORE USERS' GROUP OF SASKATCHEWAN (CUGS), Regina, Sask., Canada. It is distributed at each monthly CUGS meeting, held at 7 pm, the FIRST WEDNESDAY of every month. Meetings are usually held in the North-West Leisure Centre, at the corner of Rochdale Boulevard and Arnason St., Regina.

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 smll mailing fee for newsletters. Members are welcome to submit public domain software for inclusion in the CUGS DISK LIBRARY. Any member may purchase disks from the club library for a nominal fee. The club library looks for programs listed in such magazines as COMPUTE, GAZETTE, RUN, AHOY, TPUG, COMMODORE COMPUTING, etc. these programs are made available to members who purchase the magazines.

IN THIS ISSUE:

Meeting Place

AGENDA:

BASIC SECTION: COMPUTER "THINKING" - THE 'IF...THEN' STATEMENT by Ken D.

ED AND HIS "WEDGIES" - more Dietrich ML

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

****
**
** MULTIPLAN - SPREADSHEET EXTRAORDINAIRE!! **
**
** by Gordon Glew **
**
****

Editorial:

MICRO-MUSING

First, my sincere thanks to the regular CUGS MONITOR contributors - Richard Maze, Gordon Glew, Greg Rezansoff, Ed Dietrich, and Earl Brown.

Next, I'd LIKE to thank all you 56 or so paid-up members for the unbelievable support you've shown to your "voice" - the MONITOR - by your stream of letters, articles, reviews, and general comments.

Finally, I want to take just a moment to announce the WINNER of our great BASIC programming contest. From the pile of solutions received the winner was

****

Have a good summer but try to make YOUR CUGS MONITOR contribution happen soon!

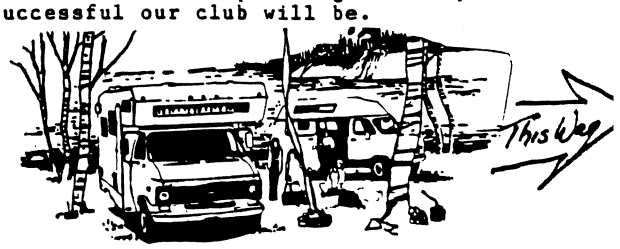
JUNE JUNKET

JUNE JUNKET - A last (summer) word from Richard.

This meeeting is the last CUGS meeting before we take our two month break in the summer. Although we won't be meeting again until September the executive is planning to be busy over the summer months. The first item we will be working on is completing the rearrangement of the DISK LIBRARY. We've completed the initial assessment of programs in our library and now are in the process of putting all the programs of similar types on their own diskettes. I am sure that this will make our collection of programs more accessible and much more meaningful. If you are interested in utility programs or arcade games or any other category, it should be much easier to narrow in on the programs you want.

A second activity that we are planning is the production of a SUMMER ISSUE of the 'Monitor'. Each executive member will be writing articles, programs, reviews, etc. for inclusion in this edition of our newsletter. If you come across anything to share with the rest of the members please pass it on to Ken. I know any and all submissions will be gratefully received.

A third activity we will be involved in is the planning of presentations for the fall. One that is at the top of our list is a presentation (or two) on communications. If there are any topics you would like to see included as presentations, or any changes you would like in the meeting format, please pass your ideas on to a member of the executive. The more input we get from you, the more successful our club will be.



Since January, our membership has almost doubled. This is a good indicator of the want and need for a club such as ours. Because of the increased enrollment some concerns about the meeting format have arisen which the executive is investigating. One involves the visibility of presentations. Presentations that involve using the computer directly to display on the screen have started to become a problem because some people just can't see the screen displays because they are seated too far away. A second problem is the extraneous noises that are part of the current facility. The executive has discussed these problems and is exploring possible solutions. If you have any ideas as to how to solve some of these problems, please pass them on to an executive member.

I hope everyone has a good summer and we will see you at our next meeting in September.

ML-ong with ED

THE INTERRUPT WEDGE

This is a technique available only to the machine language programmer. If you have ever had a program playing music in the background, a playing field scrolling constantly (often inexorably) across the screen, or the infamous smooth scrolling credit line along the top or bottom of the screen, you have seen the interrupt wedge in action.

The Interrupt is aptly named. Sixty times a second a hardware timer will cause an 'interrupt' in the processing of information by your computer. Whether you are running a machine language program, a Basic program, or simply watching the cursor flash, the 'interrupt' is constantly interrupting the proceedings. The interesting part is that it operates in the background. That is, it is invisible both to the user and to the program being run.

The interrupt is a small part (the housekeeper) of the computer's operating system. It is responsible for reading the keyboard, flashing the cursor and a myriad of other duties to keep the computer running smoothly. Because these duties take a very small fraction of the computer's time, the major portion of time is spent running the user's programs (which is only just).

A wedge is a foreign object forced into and seperating another object (roughly). Thus the wedge joins and becomes a part of the latter abused object. In this case the abused object is the operating system of the computer. The wedge is a machine language program of the user's choice (and ingenuity). The machine language 'wedge' is forced between the interrupt routine and the rest of the operating system thus making the user's program part of the operating system of the computer. This will result in the computer automatically executing the user's program every time the 'interrupt' occurs (sixty times a second).

As a rule the 'interrupt' timing is quite reliable and as in all rules there is an exception. While the timer controlling the interrupt is rock solid, the computer is quite capable of overriding (ignoring) the interrupt when facing a higher priority. The most common is the serial port. Because of the exquisite timing required for all the handshaking and bit transferring during disk operation, the computer will often 'reschedule' the interrupt to ensure error-free (and abysmally slow) data transfer. If the wedge controls graphics, they will jitter and stutter during disk operation. What happens to music I refuse to discuss. It is usually wise to disable the wedge during disk operation and enable it afterward.

TECHNICAL STUFF & TIPS

The Interrupt vector is stored in standard Hi-byte Lo-byte format in locations $0314 and $0315. This points to the interrupt handling routine at $EA31.

To insert a wedge;
1. Use the SEI (set interrupt flag) command to inhibit the interrupt while playing with the interrupt vector.
2. Place the starting location of your machine language program in $0314 and $0315 (Hi-byte Lo-byte format).
3. Use the CLI (clear interrupt flag) command to re-eanable the interrupt.
4. Say a short prayer and see what happens.

Unless your wedge program is capable of handling the housekeeping chores it should end with a jump to $EA31 to prevent the computer from crashing.

For the advanced programmer: Whenever timer A of the 6526 Complex Interface Adapter (CIA) #1 counts to zero, bit zero of location $DC0D is set. This normally (but not always) generates an interrupt. It depends on the contents of the Interrupt Mask Register (location $D01A). Bit seven (normal operation) enables the timer interrupt. Bit three enables the light pen interrupt, bit two is sprite collisions, bit one is sprite to background collision, bit zero enables raster interrupts. If more than one interrupt is enabled, your wedge program will have to check location $D019 (Video Interface Chip Interrupt Flag Register) to see what event caused the interrupt so that your program may handle it. The flag in location $D019 must then be cleared before exiting your interrupt handling routine.

INTERRUPT-DRIVEN JOYSTICK

This program uses the 'interrupt wedge' technique. It allows the user to control Sprite #0 with a joystick inserted in Port #1. This program will only move the sprite. The user will have to draw and switch on Sprite #0 from within his own program. The fire button is not checked and your own routines for sprite collisions will have to be written. This can all be done fairly simply from Basic.

The true beauty of this routine is that it modifies the computer's operating system. Once the 'wedge' is enabled, the joystick is checked sixty times per second and Sprite #1 is moved automatically. Because the computer is technically not running a program, the user can then run his own Basic or ML program and not have to read the joystick or move the sprite. The computer will do this automatically.

Because disk operation interferes with the interrupt timing, it is wise to disable the 'wedge' prior to accessing the disk drive. The 'wedge' can then be re-enabled following any disk operation.

To enable the 'interrupt wedge' type SYS 49152 and press RETURN

To switch on Sprite #0, type POKE 53269,1 and press RETURN.

To disable the 'wedge' type SYS 49292 and press RETURN.

To enter the program, use XMON64C(50135) available on tne U1 disk from the club library. Ignore the line numbers to the left of the ML commands, they are there for reference only.

As the program will start at $C000 or 49152 (decimal) the first line should be

.A C000 SEI

After this, type in the rest of the program following the prompts.

```
 1. SEI          24. BNE $C03A   47. BEQ $C07C
 2. LDA #$0D     25. JSR $C065   48. LDA $D000
 3. STA $0314    26. JMP $EA31   49. CMP #$56
 4. LDA #$C0     27. DEC $D001   50. BNE $C088
 5. STA $0315    28. RTS         51. LDA #$00
 6. CLI          29. INC $D001   52. STA $D010
 7. RTS          30. RTS         53. STA $D000
 8. LDA $DC00    31. LDA $D000   54. RTS
 9. AND #$0F     32. BNE $C061   55. LDA $D000
10. STA $C099    33. LDA $D010   56. CMP #$FF
11. AND #$01     34. AND #$01    57. BNE $C088
12. BNE $C01C    35. BNE $C05C   58. LDA #$01
13. JSR $C03D    36. LDA #$01    59. STA $D010
14. LDA $C099    37. STA $D010   60. INC $D000
15. AND #$02     38. LDA #$56    61. RTS
16. BNE $C026    39. STA $D000   62. SEI
17. JSR $C041    40. RTS         63. LDA #$31
18. LDA $C099    41. LDA #$00    64. STA $0314
19. AND #$04     42. STA $D010   65. LDA #$EA
20. BNE $C030    43. DEC $D000   66. STA $0315
21. JSR $C045    44. RTS         67. CLI
22. LDA $C099    45. LDA $D010   68. RTS
23. AND #$08     46. AND #$01
```

Memory Locations of Interest:

$0314 & $0315 -Contain the hardware interrupt vector ($EA31).
$DC00 -The lower four bits (0-3) indicate the joystick position
$D000 -Sprite #0 X-position on screen
$D001 -Sprite #0 Y-position on screen
$D010 -Most Significant Bit of X-position of sprites

A Brief Explanation

Lines 1-7 enable or insert the wedge.
Lines 62-68 disable the wedge.
Lines 8-26 inclusive comprise the WEDGE that is inserted into the computer's operating system.
Lines 8-10 read the joystick, mask out the necessary bits and store the result in a temporary location.
Lines 12-25 check the position of the joystick and move Sprite #0 accordingly.
Line 26 is the last line of our routine and jumps to location $EA31, the normal interrupt routine which performs the housekeeping tasks and then returns control to the main program (if one is running).
The following are subroutines called from within our wedge to move Sprite #0.
Lines 27-28 (subroutine) move Sprite #0 up one pixel.
Lines 29-30 (subroutine) move Sprite #0 down one pixel.
Lines 31-44 (subroutine) move Sprite #0 left one pixel.
Lines 45-61 (subroutine) move Sprite #0 right one pixel.

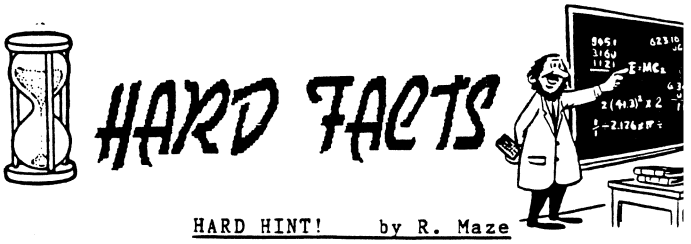
The program can be saved to disk with the command,

.S "0:filename",08,C000,C099

BUMMING

Summer holidays soon. Time .... Glorious time! Time to plant, a time to reap, a time to laugh, a time to... (ahem, pardon me). If your wife is anything like mine, the MINUTE she suspects I've got 2 free consecutive minutes, she finds 6 hours work for me to do around the house. Here it is!! Your big chance! The EXCUSE YOU'VE ALL BEEN WAITING FOR!!!

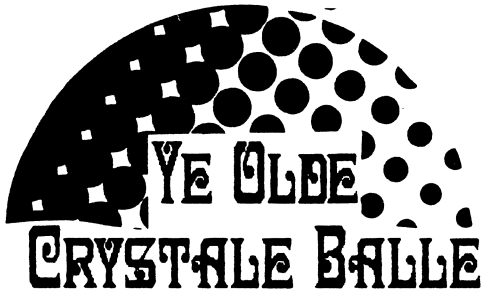
Simply inform your wife (or boss, if you want a real chuckle) that YOU'RE BUSY WRITING THAT MUCH REQUESTED ARTICLE FOR YOUR FAVORITE COMPUTER RAG - THE MONITOR. I'll accept virtually any format - PPaperclip, Paperback Writer, WordPro, Easyscript, Speedscript, pencil, pen, crayon, tempra paint, lipstick.... Anyhow, you're news, views and queries are much appreciated and printed (THAT'S a promise). Use part of YOUR summer to help make MY summer easier. Muchas gracias, Dobre, etc., etc.



HARD HINT! by R. Maze

A harware modification to the disk drive.
THE FOLLOWING MODIFICATION WILL VOID THE WARRANTY ON YOUR DISK DRIVE

Should you wish to format and write on the back side of a disk, without having to punch a write-protect hole the following is a neat way to achieve it. Open up your disk drive. Take a look at the long, flat, white plug which has a number of wires coming from it. See the first two wires in this plug - the two closest to the front of the drive? These wires lead down to the write protect sensor. IF you install a micro-switch that will connect these two wires when flipped ... you can format and write on the back side of any disk without punching any holes in the diskette. With the switch off, the drive will function normally.



Keep your eyes peeled, or at least partially open, for posters announcing our SEPTEMBER MEETING. Nex year, we'll be spending a fair bit of time talking about MODEMS AND ON-LINE COMPUTING - so, any questions, comments, etc.???

As you'll read elsewhere this issue, our new CUGS DISK LIBRARY CATALOGUE will be MAILED to every paid-up member. If you think that's you, and you don't have a copy by late AUGUST, call EARL BROWN or KEN DANYLCZUK.

Rest up this summer, we've a great busy year come September!!

SIR RICHARD'S BASIC

SIR RICHARD'S BASIC - by Richard Maze

An often difficult-to-understand aspect of programming in BASIC is the array. This is the first two explorations of arrays. This month, I will examine the array and how to set up an array in a BASIC program. Next month, I will give some examples of using arrays.

An array permits the storing of a number of variables using one variable name. Simply, an array can be visualized as a shelving unit. The entire shelving unit is called by one variable name (NU). Each shelf in the shelving unit is numbered and and can hold one item. For example, an array called 'NU' could have 25 shelves and be able to store 25 numbers. Each number would be accessed using the name of the array (NU) followed by the number of the shelf it is on in parentheses. NU(16) would refer to the number stored on shelf 16 of the array called NU.

An array is set up using the DIM statement. The DIM statement causes the computer to set aside memory to accomodate the array. For example: DIM NU(25) would set aside space for an array called NU to have 25 shelves. (NOTE: this is actually 26 shelves as a shelf 0 exists as well.) More than one array can be set up in a program by giving the names of the arrays after a DIM statement with each separated from the next by a comma. For example: DIM NU(25), NA$(25) would set up two arrays each with 26 shelves (don't forget the 0 shelf).

Arrays follow all the rules for variables - there are 3 kinds (floating point, integer and string) of arrays that may exist. The same rules for assigning values to variables apply to arrays. An array name is stored differently than a normal variable name, however, so it is possible to use the same variable name for a normal variable [NU] as well as for an array [NU(25)].

Arrays can be filled with data in much the same way as normal variables - by direct assignment, assigning results of calculations, from INPUT or GET statements, from DATA statements, from files. The only difference is the ease of use of arrays where one variable name is used for all data as compared to regular variables which must have a separate variable name for each data item. As a result, arrays can usually be filled with data by using a FOR...NEXT loop. The following program segments show how an array can be filled with data.

```
100 REM EXAMPLE USING INPUT
110 DIM NA$(20),NU(20):REM 2 ARRAYS EACH WITH 21
SHELVES
120 FOR X = 1 TO 20:REM IGNORE SHELF 0
130 :   PRINT "SHELF" X;": ";
140 :   PRINT TAB(12)"ENTER NAME";:INPUT NA$(X)
150 :   PRINT TAB(12)"ENTER NUMBER";:INPUT NU(X)
160 NEXT X
170 PRINT "CLR":REM CLEAR SCREEN
180 PRINT "SHELF","NAME",,"NUMBER"
190 FOR X = 1 TO 20
200 :   PRINT X,NA$(X);TAB(32)NU(X)
210 NEXT X
```

Enter the following lines as replacements for the lines indicated in the program above.

```
100 REM EXAMPLE USING READ/DATA
130 :   READ NA$(X),NU(X)
DELETE LINES 140,150
500 DATA AAAAA,20,BBBBB,34,CCCCC,16,
DDDDD,19,EEEEE,56,FFFFF,88,GGGGG,44
510 DATA HHHHH,99,IIIII,56,JJJJJ,24,
KKKKK,79,LLLLL,59,MMMMM,81,NNNNN,73
520 DATA OOOOO,62,PPPPP,75,QQQQQ,37,
RRRRR,76,SSSSS,40,TTTTT,62
```

The following program simulates the rolling of a pair of dice 500 times. Try doing this without arrays!

```
100 DIM COUNT(12)
110 FOR X = 0 TO 12:COUNT(X) = 0:NEXT
120 REM LINE 110 NOT REALLY NECESSARY AS ARRAY
SHOULD BE INITIALIZED TO 0
130 PRINT"ROLLING - PLEASE WAIT!"
140 FOR X = 1 TO 500
150 :   Y = INT(11*RND(1)+2):REM CREATE THROW
BETWEEN 2 AND 12
160 :   COUNT(Y)=COUNT(Y)+1:INCREMENT THROW COUNTER
170 NEXT
180 PRINT"ROLL","NUMBER"
190 FOR X = 2 TO 12
200 :   PRINT X,COUNT(X)
210 NEXT
220 END
```

Try changing the above for 3 dice.
Next month, I'll examine TWO-DIMENSIONAL ARRAYS as well as a few more uses of arrays in programs.

Diskette-iquette

by Earl Brown

Here it is already June and the summer break is almost here. Our Disk Library is just about in its final stages of a new conversion. Our software will be separated up into various catagories for our C-64 programs. These catagories will be:

1. COMPUTER UTILITIES
2. DISK UTILITIES
3. PRINTER UTILITIES
4. ARCADE TYPE GAMES
5. ADVENTURE & OTHER GAMES
6. COMMUNICATIONS
7. GRAPHICS
8. SOUND
9. BUSINESS
10. GENERAL

Since this is quite an undertaking, we don't believe for a minute, that this will be the final step in the process. We are bound to miss some duplicates, similar programs, the odd one perhaps that do not completely work, and programs placed into the wrong catagory. None the less, the entire library, should be better organized for our members' use. A catalogue of the new library will be mailed to each paid-up member some time during the summer. Although the club will not be meeting during July and August, any members who wants to reach me during this time may simply give me a phone call, and I will see what I can do.

GAZETTE disks 16 & 17 are ready for members to purchase this month, as well as TPUG for April '87. I've added Disk # 41 with a few more educational programs for the C-64 and Pet computers. And finally, the Input Disks we ordered have just arrived. I didn't have time to back them up but if you would like copies of any of the disks, let me know and I'll oblige. There wasn't time to run any of these programs yet so I cannot comment on them except that at least 2 programs on the C-128 Disk #2 does not completely work. Thanks to Real Charron for that bit of advise as well as all the AHOY! programs from March through December of 1986. These programs will be made available on future Cugs Disks. They too, were just received and were not available soon enough to catalogue.

P.S. Thanks to Ken Danylczuk (our Monitor Editor) for a copy of TRANSACTOR's TransBASIC Disk #1. This disk will also become a part of our library.

P.P.S. SYS 65341 warm starts the 128 mode of the C-128 computer.

NEW CUGS DISK LIBRARY ADDITIONS;

DISK# 16 64 GAZETTE

4	.DIR
1	3.<--------->
4	BORDER 1-4
8	TINYTERM
54	WEATHER PROPHET
5	QUICKSCAN
25	MAZE-MANIA
3	VIC EMULATOR
8	VIC.49152
19	PRINTER WEDGE
2	CHAR SET 1
4	CHAR SET 2
9	CHAR SET 3
5	CHAR SET 4
1	3.<---------->
2	SYNTAX CHECKER
4	SYN CHECK.49152
2	TURNABOUT
10	TURNABOUT.49152
4	JOYSTICK READER
11	KALEIDOSCOPE
11	ATOM SHOOT
44	THE FARM GAME
2	X BASIC
12	X BASIC.49152
2	GRADEBOOK
13	GRADEBOOK.49152
4	GRADEBOOK OPTION
1	3.<---------->
7	GOBBLEDYGOOK
4	PREVIEW 80
5	PREVIEW.52000
4	SCREEN CUSTOMIZR
4	AUTO FILE.BOOT
4	AUTO FILE
11	CHICKEN CATCHER
3	DISK ENCODE.BOOT
7	DISK ENCODER
2	MUSIC MAKER
2	MUSIC.49152
27	POWER POKER
15	BACKGAMMON
9	DIGI-CLOCK
1	3.<---------->
9	AUTO DATE/TIME
2	FILE ARCHIVER
2	ARCHIVER.49152
10	SOUNDPIX
5	LIST PAGER GEN.
4	LIST PAGER
1	LIST PAGER.OBJ
13	HOME BUDGET/W/VS
10	DISK PROTECT/UN
22	DRAGON'S DEN
2	WHIRLYBIRD
13	WHIRLYBIRD.49152
2	QUICKCHANGE
9	QUICK.49152
8	BANNERS/1525
8	BANNERS/1526
7	BANNERS/1525/VIC
7	BANNERS/1526/VIC
21	CONSTRUCTION SET
9	HOUSE
9	CREATURE
9	SHAPES
3	TRACKMOUSE
15	DICTIONARY MGR
11	SPEEDCHECK
6	A - Z

DISK# 17 64 GAZETTE

4	.DIR
1	4.<->
30	DISK DISASSEMBLR
2	A COMPILER
24	SPRINT.32768
5	DOODLER DEMO
2	SPACE ARENA
16	ARENA.49152
4	FAST ASSEMBLER
11	ASSEMBLER
58	ASSEMBLER.SOURCE
3	PRINT MAKER
2	PRINT.49152X
3	PRINT MAKER/VIC
2	PRINT.7168X
27	FACE-OFF
18	NEW MLX
4	REM HIGHLIGHTER
3	HIGHLIGHTER/128
3	HIGHLIGHTER/VIC
2	OFF SCREEN TRACE
4	TRACE.34816
22	BASIC BACKUP
6	BASIC WINDOWS
1	4.<-->
15	CUSTOM LABELS
2	DISK EDITOR
8	DISK ED.12000
3	KICKER
7	KICKER.OBJ
18	LEXITRON
18	LEXITRON/128
7	NEW PROOFREADER
8	BLINK MODE
8	BLINK MODE/VIC
2	BLINK.OBJ
9	BM-DEMO 1
6	BM-DEMO 2
2	BLINK.OBJ/VIC
8	BM-DEMO 1/VIC
6	BM-DEMO 2/VIC
3	SNAPSHOT
9	SNAPSHOT.49444
4	SNAPSHOT DEMO
5	MINI-FILER
9	MINI-FILER.OBJ
1	4.<--->
24	SHIFTER
7	TELECONVERTER
7	SCREEN DUMP/VIC
15	CATALOGER
8	CLAVIER
15	COORDINATOR.LDR
2	OBJECT MAKER
1	A COORD.OBJ
3	B COORD.OBJ
21	COORD DEMOS
14	CONSTRUCTION KIT
9	KEY TO JOY CONV
8	WORM DEMO
7	SURVIVOR
11	SURVIVOR.OBJ
2	AUTOBOOT/64 4
6	AUTOBOOT/128 3
6	AUTOBOOT/128 2
6	AUTOBOOT/128 1
1	4.<---->
18	DIRECTORY FILER
5	WINDOW DEMO/128
5	WINDW SAVE40/128
5	WINDW SAVE80/128
11	INPUT.OBJ GEN
2	INPUT.OBJ
3	INPUT DEMO
2	DUNK
8	DUNK.49152
3	TURBO COPY
11	TURBOCOPY.49152
3	COORD LOAD & DIS

DISK# 41 CBM PGMS

4	CBM 4032 V2.1
44	ANDROID NIM.C2
80	BOUYANCY.C2
36	BOWLING.C2
30	BROWNIAN.C2
48	COMMODITY.C2
24	BOTTLECAPS.C2
27	CHEM CALC.C2
26	CHEMIST.C2
55	DRACULA.C2
38	DRAGON ISLAND.C2
39	DRAGON MAZE.C2
65	DRIVER ED.C2
30	DROIDS.C2
69	ELECTRO MAG 2.C2
45	FACES TO MAKE.C2

DISK# AJ TPUG AP87

30	AUTOBOOT
6	PRINTBOOTDATA
11	AUTODOC
61	EA-EDIT/ASM.PROG
103	EA-EDIT/ASM.DOC1
94	EA-EDIT/ASM.DOC2
119	TAX EH!D 2
13	EH'D AID
56	NEW TREK
21	TREK INSTRUCTION
51	WHEEL
44	TAX 86
17	SOLITARE
6	FAST FORMAT
7	FAST FORMAT.D
5	BOOT.DATA
6	SEQ READER

DISK# T1 TRANSBASIC

2	TRANSBASIC
7	TB/USE.OBJ
13	SYMASS 3.10
15	TB/KERNEL
1	-- UTILITIES --
10	SUPERMON64.V1
41	UNASSEMBLER
1	SEQ TO PAL
8	C64 TINY AID LDR
17	CHECK KEYWORDS
4	STATEMENT LIST
1	FUNCTION LIST
33	CHECK LABELS
3	STRIPPER
7	DATAFIER
1	-- TB MODULES --
24	ADD
32	USE
	SCREEN THINGS
4	DOKE & DEEK
3	BIT TWIDDLERS
4	CHECK & AWAIT
3	KEYWORDS
4	CURSOR POSITION
7	SET SPRITES
5	WITHIN
4	READ SPRITES
5	STRIP & CLEAN
20	SCROLLS
10	LABELS
5	TOKEN & VAR
8	INSTRING
7	PLACE
7	ARCFUNCTIONS
3	PRINTAT
18	SOUND THINGS
12	MOVE & FILL
11	DOS SUPPORT
5	LINE CALC
3	BEEP
26	PRG MANAGEMENT
4	COMPUTED CMDS
5	RANDOM
5	PHRASE SPLITTERS
2	OLD
7	INPN & INPA
5	SELECT
46	MC GRAPHICS
4	INLINE
3	DELAY
7	SLIDE
2	MAKE
3	CENTRE
11	VOCAB MANAGER
10	STRING SYNTHESIS

zzzzZIP by Gordon Glew PERSssss

IF TIP - The mathematical expression between IF an THEN determines wheather the rest of an IF ststement will be executed. When the expression is false, the rest of the line is skipped.

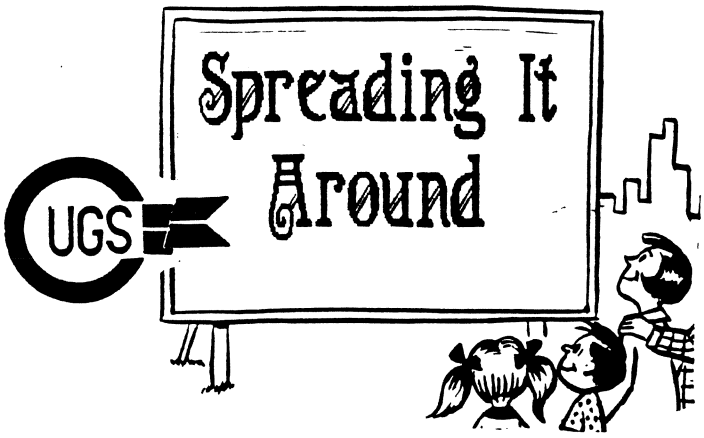
You can use this feature to save execution time. Rather than using a statement like: 100 IF X=1 AND Y=2 THEN PRINT Z. it is much faster to write 100 I X=1 THEN IF Y=2 THEN PRINT Z.

In the first case, X=1 AND Y=2 must be evaluated before any line skip decision is made. In the second, as soon as X=1 is evaluated as false, everything else is skipped. The result is faster execution whenever X=1 is false.

IF..THEN..ELSE - Unfortunately, Commodore Basic doesn't have this useful construction, which allow you to redirect the program if the IF statement fails. You can use the ON...GOTO statement to give a similar effect, as in this example:

```
400 GETA$:IFA$=""THEN 400
500 ON((A$="Y")+2)GOTO 600:ON((A$="N")+2)
      GOTO 700:GOTO 400
600 PRINT"YES":END
700 PRINT"NO":END
```

Note how the two tests have been put on the same program line.



SEARCHING FOR A SPREADSHEET - by Gordon Glew

In the past it was a rather simple matter to select a spreadsheet for the C64. With only a couple to choose from, it was hard to go wrong. Now there are a number of spreadsheets to compare, with capabilities that put the C64 in the same league as Apple and IBM. Along with increased spreadsheet selection and sophistication comes the difficulty of chosing the right one for your needs. A little background on spreadsheets before you start looking may help prevent that sick feeling of finding the perfect program after you purchase the wrong one!

Essentiaally, all spreadsheets set out to let you create charts of names and numbers and perform mathematical operations on those numbers. Most begin by displaying a similar screen. A row of numbers along the top corresponds to columns on the screen. Along the left edge alphabetic characters correspond to each row on the screen. The intersection of a column and a row is refered to as a CELL and can be highlighted with a movable cursor. Although the C64 displays a 40 character by 25 character screen, you can develop a spreadsheet that is much larger than that. To view other parts of the spreadsheet, move the cursor to the extreme edge and the entire sheet appears to move beneath the screen. It is a little like looking through a window. You never see more than a 40 by 25 character section at one time, but you can move the screen to view other sections of the sheet below it.

Using the keyboard the cursor can move in any direction from cell to cell. Once the cursor is positioned in a cell, data can be keyed in. Into a cell you can enter a name, number or formula. You could create a column of names corresponding to the expenses you have with your automobile. The column might include GAS, INSURANCE, REPAIRS, and PARKING. You could then enter into an ajacent column the expected expense of each category: $45, $30, $10, $14. Not impressed yet? Well, the real power of a spreadsheet is in the formulae. Positioning the cursor in a cell below the list of expenses, directly below $45, you could enter a formula that totals the entire column. On one spreadsheet the entry would appear as Sum(B1-B4) where B1-B4 references the CELLS comprising the column. What suddenly appears in the bottom cell is not a formula itself but the result. The formula will disappear from view when you move the cursor but can always be recalled if necessary.

Now for the real magic of spreadsheets. Move the cusor to one of the values, say the gas expense of $35, and change it. The total value at the bottom automatically changes, too. Change any of the values and, each time, the total is recalculated and displayed automatically. Applications of spreadsheets reach far beyond budgets and accounting. You can use the same program to creat a sheet, average your grades in school, monitor th fluctuations in the stock market, or track the sales of your staff. Once a spreadsheet has been developed, repetitive calculations in engineering or chemistry are as easy as keying in variables. What impact on your savings over the next month will commuting to work on the bus have? Reduce th parking expense to $3, and the gas expense to $10, and instantly a new total appears.

You could have used a calculator, crossing out old values and inserting new ones, then recalculating each time, but with a long list of numbers and formulae that process becomes tedious and is prone to error. The convenience of changing original values to obtain new totals makes trial and error problems a breeze. Trial and error problems are very popular among financial planners, and they ar one of the reasons spreadsheets are so popular in the business community.

Armed with a little background on spreadsheets, what features should you look at in comparing them

CONFIGURABLITY

If you've read much about software or hardware, th word configurablity has always been a menace. It is one of those "buzz words" that comes as sort of a warning that your peripheral or program may not work the way you want. Configurablity is actually a reference to the flexibilty and limitions of a device or program. In selecting a spreadsheet, yo want to be aware of its configurablity. You are not going to know exactly how many rows or colums you will need on applications you haven't even thought of yet, but compare the maximum for each spreadsheet. The maximum number of cells is anothe statistic. If the software advertises a maximun of 256 rows, 100 colums, and 1000 cells, that does no mean you can have both 256 rows AND 100 columns. That would work out to 25,600 cells. It means you can have up to 256 rows OR 100 columns as long as you do NOT exceed 1000 cells.

Look for a spreadsheet that states it is configure for your type of printer. Some are configured only for the 1525; some other printers will give unpredictable results. If it states it works with any properly configured printer you are probably safe. You're not going to get a stronger statemen out of a software vendor.

Some spreadsheets produce graphs that can be outpu to the printer. If you are looking for graphs be especially careful that the spreadsheet can be configured for your printer. The width of a colum displayed on the screen varies with the different spreadsheets available. Most spreadsheets allow you to change the width up to some maximum value for the entire sheet.

EDITING CAPABILITIES

All spreadsheets allow you to enter values into a cell and make corrections if necessary. But there are some handy little features that you will invariably wish you had that are available on some of the better programs. Row and column copy commands are a must. If the column under January has 100 entries and is identical to the one that you must enter for February, about 80 entries down

you're going to wish you had a column copy command: one that would copy all the entries from one column another identically. If you have formulae in the column that make reference to the first column, you may want to have a relative change option in the copy command. This feature allows you to copy a column or row and automatically change any formula entry that references the column. If you are copying column 4 to column 6 and a formula references column 4, you may want it changed to 6 to reflect its new location. Some spreadsheets allow you to insert a column or row between two adjacent ones. This is a little tricky. If you insert a column between C and D what is the spreadsheet going to call it? Don't worry; they each work a little differently, but it can be done.

If you are entering lists of numbers under a row of names, as you move down the list the names will eventually scroll out of view. One useful feature is the ability to fix titles. This feature will allow you to hold a row or column in view regardless of where you are viewing the spreadsheet.

CALCULATIONS

All spreadsheets will allow you to add, subtract, multiply, and divide. Most go far beyond that, performing mathematiacal operations I can't even pronounce, You generally will pay for those advanced features and never have an opportunity to use them. On the other hand, you may be one of those that just find it comforting knowing they are there. Look for a list of the calulations available on a spreadsheet before you buy. Some spreadsheets allow formulas that include IF...THEN commands. If (A1)>$10 THEN $40 ELSE $20 could be a typical formula. It would yield $40 if the value in cell A1 turned out to be greater than $10, and $20 if the value in cell A1 were less than or equal to $40. This is a very useful type of calculation in many business applications. If you have discounts for certain quantities sold, or a bonus on certain sales, this feature would facilitate the entries into a spreadsheet.

ADVANCED FEATURES

Remember, generally, the more sophisticated spreadsheets require more effort in learning and application. Try to find a balance between versatility and ease of use. There are some advanced features that you may or may not want. Remember, the screen only displays 40 by 25 characters of the spreadsheet at a time, some spreadsheets allow you to split the screen to view two seperate sections together. If you split the screen vertically, you can view two sections each 20 characters wide and move each section independent of the other.

One very powerful feature found in better software is a linking capability. Some spreadsheets allow you to create formulae that reference cells in other spresdsheets. You could create a home budget spreadsheet that would automatically look up values on your auto expense spreadsheet. Some spreadsheets allow you to display and print graphical representations of lists of numbers you create. Remember, different vendores have their own ideas about what a graph is. It may be anthing from a row of asterisks to a high res multicolor display. Any elaborate printout of graphs requires that your spreadsheet be designed to work specifically with your printer.

Along side is a chart comparing a few spreadsheets for the 64:

SPREADSHEET FEATURES

	RELATIVE COPY	IF THEN COMMAND	R/C INSERT	REJUSTIFY	SPLIT SCREEN	ADJ.COL WIDTH	LOGIC OPERATORS	FIX TITLES	GRAPHS	SORT	SEARCH	LINK	MAX COL. WIDTH	MAX COLS.	MAX CELLS	MAX ROWS	PRICE
BCALC	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	7	99	99	3600	36	$44.95
BUSICALC	YES	NO	YES	NO	YES	YES	NO	NO	NO	NO	NO	10	99	99	1100	99	$69.00
CALC RESULT ADV	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	10	63	1000	63000	999	$149.95
CALC RESULT EASY	YES	YES	YES	NO	YES	NO	YES	YES	YES	NO	NO	10	63	1000	63000	999	$79.95
ESP>CALC	NO	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	0	640	2000	64000	640	$47.50
MULTIPLAN	YES	YES	YES	YES	YES	YES	YES	YES	NO	NO	NO	10	63	M	200	M	$99.00
OMNICALC	YES	YES	YES	NO	YES	NO	YES	YES	NO	NO	YES	20	63	M	200	M	$49.00
PRACTICALC	YES	NO	YES	NO	YES	NO	YES	YES	YES	NO	NO	20	63	100	2000	M	$60.00
SPREAD ASSISTANT	YES	NO	YES	NO	YES	NO	NO	NO	YES	NO	NO	20	62	M	200	M	$125.00

MULTIPLAN - MSDN

MULTIPLAN REFERENCE GUIDE by Gordon Glew

MOVE THE CELL POINTER

Action	Key to use
Up	Up arrow
Down	Down arrow
Left	Left arrow
Right	Right arrow
Next window	CTRL W
Next unlocked cell	CTRL F or INST

SCROLL THE WINDOW

Action	Key to use
Page Up	F5 or CTRL R up arrow
Page Down	CTRL R down arrow
Page Left	F7 or CTRL R left arrow
Page Right	CTRL R Right arrow
Home	Home or CTRL S
End	CLR or CTRL Z

SELECT and EXECUTE COMMANDS

Action	Key to use
Cancel	CTRL C or RUN/STOP
Do this command	RETURN
Select next item on Menu	SPACE BAR
Select previous item on Menu	DEL
Tab to field on Command	F1 or CTRL I or CTRL A
Help	?
Recalculate	!

EDIT CELLS and COMMANDS

Action	Key to use
Delete	F3 or CTRL Y
Character Left	F4
Character Right	F6
Word Left	F2
Word Right	F8
Reference	@
Backspace	Del

LOAN ANALYZER for use with MULTIPLAN

(or a similar spreadsheet)

1 With the cursor in cell R1C1
  Type the name of the spreadsheet.
  Type A for Alpha Command
  Type LOAN ANALYZER
  Type RETURN

2 Move the cursor to R3C1 and begin entering labels for the entries
  Type A for Alpha command
  Type PRINCIPAL
  Type down arrow
  Type INTEREST RATE
  Typr down arrow
  Type TERM
  Type down arrow twice
  Type MONTHLY PAYMENT
  Type RETURN
  None of these lables are shown in whole on the screen so make the column wider
3 Make the column wider so it will display all the text
  Type F   for format
  Type W   for width
  Type 15   for number of characters
  Type RETURN

4 Now make the lables in these columns flush right by moving the cell pointer to R1C1 (using the home key).
  Type F   for format
  Type C   for cells
  Type :   beginning of block of cells
  Type R7C1 end of block of cells
  Type F1   tab left once to align
  Type R   align entries right
  Type RETURN

ENTER THE FORMULA
1 Move the cursor to cell R7C2
  Type (to begin formula
  Type with cursor up four times
  Type *
  Type with cursor up three times
  Type / for divide
  Type 12)/(1-(1+
  Type up cursor three times
  Type /12)up arrow(-
  Type up cursor twice
  Type *12))
  Type RETURN

A LOOK AT A LOAN
  You are shopping for a car. The price of the car is $10000.00 and your going to put $2000.00 down. You want to loan for $8000.00. A loan is available at 12% interest with 3 years to pay

  1 In cell R3C2
    Type 8000
    Type cursor down
    With cursor in R4C2
    Type .12
    Move to cell R5C2
    Type 3
    Type RETURN

  MAKING SENSE OF THE DOLLARS
    Move the cursor to R3C2
    Type F   for format
    Type C   for cell
    Type F1 twice
    Type $
    Type F1
    Type 2
    Type RETURN

  WIDDEN COLUMN
    Type F   for format
    Type W   for width
    Type 12 for number of characters
    Type RETURN

  Next move the cursor to R4C2
    Type F
    Type C
    Type F1 twice
    Type %
    Type RETURN

  Next move to R7C2
    Type F
    Type C
    Type F1 twice
    Type $
    Type F1
    Type 2
    Type RETURN

  If you want to compare your options do the following:
  1 Move to R3C3
    Type F for format
    Type W for width
    Type 12
    Type F1 twice
    Type 6
    Type RETURN

  Move the cursor to cell R3C2 ($8000.00)
    Type C   for copy
    Type F   for from
    Type R7C2 set of cells to copy
    Type F1
    Type R3C3:R3C6
    Type RETURN