CLEVELAND AREA TI-994/A USER GROUPS NEWSLETTER

SEPTEMBER, 1990

OFFICERS	NORTHCOAST	TI-CHIPS
CO-PRESIDENT	STEVE BAGSTED 1-933-5977	MATT ANDEL 676-9759
CO-PRESIDENT	BOB KAGY 1-255-2609	GLENN BERNASEK 238-6335
TREASURER	FRANK JENKINS 283-8526	LIN SHAW 235-3912
MEMBERSHIP	CHUCK POULIN 731-6473 361 E. 280TH ST. EUCLID, OH 44132	JOHN PARKEN 331-2830 4172 W. 217TH ST. FAIRVIEW PARK, OH 44126
SECRETARY	CHUCK POULIN 731-6473	DENNIS LIKENS 842-9627
LIBRARY(DISK)	MARTIN SMOLEY 1-257-1661	HARRY HOFFMAN 631-2354
(TAPE & MODS)	FRANK JENKINS 283-8526	JOHN PARKEN 331-2830
(HARD COPY)	DICK ALDEN 1-352-9172	

MEETING DATES	
NORTHCOAST 1:30 P.M.	TI-CHIPS 10 A.M.
EUCLIDIAN ROOM	N.ROYALTON LIBRARY
EUCLID SQUARE MALL	STATE RD & RT 82
THIRD SATURDAY	THIRD SATURDAY
SEPTEMBER 15, 1990	
OCTOBER 20, 1990	
NOVEMBER 17, 1990	
DECEMBER 15, 1990	
JANUARY 19, 1991	

REMEMBER, ONLY THREE MORE ISSUES AND YOU WILL NEED A NEW NEWSLETTER EDITOR. WE HAVEN'T EXACTLY BEEN OVERWHELMED WITH VOLUNTEERS. BEING NEWSLETTER EDITOR MEANS PUTTING IT TOGETHER, NOT HAVING TO WRITE IT, SO IT'S NOT AS DIFFICULT AS YOU MIGHT IMAGINE. WE ARE VERY LUCKY TO HAVE PEOPLE HERE WHO CONTRIBUTE REGULARLY WITH THEIR CONTRIBUTIONS SUBMITTED ON DISK OR "READY TO PRINT".

I HOPE TO ONE OF THOSE WHO WILL SUBMIT REGULAR OR SEMI-REGULAR ARTICLES. AS YOU KNOW, I LIKE GRAPHICS PROGRAMS, BUT TIMEWISE JUST HAVEN'T BEEN ABLE TO GET INTO THEM THE WAY I WOULD LIKE. AS I DO, I WOULD LIKE TO SHARE MY EXPERIENCES WITH YOU AND THAT IS ONE OF THE MAIN REASONS I AM GIVING UP THE NEWSLETTER JOB. FOR SOME REASON, A SHORTAGE OF TIME SEEMS TO BE MY BIGGEST PROBLEM. I KNOW A LOT OF YOU ARE IN THE SAME SITUATION, BUT HOPE SOMEONE WILL COME FORWARD.

I THINK WE ALL GOT MIXED UP WITH THE SUMMER VACATION AND NO NEWSLETTER LAST MONTH. FOR INSTANCE, I HAVE DENNIS LIKEN'S REPORT FOR JULY AND NONE FOR AUGUST. I DIDN'T HEAR FROM ANYONE FROM NORTHCOAST FOR AUGUST, SO WROTE A SMALL RECAP MYSELF.

BOTH THE NC AND CHIPS REPORTS MENTION NEW DISKS FOR THE LIBRARY WHICH WE HAVE CHOSEN NOT TO PRINT IN THE NEWSLETTER. IF YOU WEREN'T AT THE LAST TWO MEETINGS, GET A COPY FROM YOUR RESPECTIVE LIBRARIAN. THERE WILL BE ANOTHER GROUP COMING IN SEPTEMBER AND POSSIBLY EVEN INTO OCTOBER. BESIDES THE DISKS FROM LIMA, I HAVE SEVERAL WHICH WES RICHARDSON GOT ON A TRIP TO PORTLAND AND SOME I HAVE DOWNLOADED FROM GENIE. THUS WE SHOULDN'T RUN OUT OF NEW SOFTWARE IN THE NEAR FUTURE.

CHIP'S REPORT MENTIONS THAT WE HAVE TIPS V1.6. HARRY GAVE ME A COPY AT THE AUGUST MEETING. I GAVE IT A QUICK TOUR AND YOU SHOULD FIND THAT OUR BIGGEST COMPAINT ABOUT THIS PROGRAM "SLOWNESS" IS A THING OF THE PAST. IT PERFORMS VERY NICELY AND QUICKLY. AGAIN, WE SHOULD ALL REMEMBER TO THANK RON WOLCOTT FOR THIS CONTRIBUTION TO OUR TI WORLD.

WHEN YOU SEE MARTY SMOLEY'S TI-BASE COLUMN THIS MONTH, YOU WILL PROBABLY THINK HE HAD IT PRINTED PROFESSIONALLY. HE DID, WITH FUNNELWEB AND A LASER PRINTER. HE HAS A NEW CANON LASER PRINTER WITH SCALEABLE FONTS. HE IS SCROUNGING UP EVERY CANON LASER PRINTER DRIVER HE CAN FIND TO GET CODES TO USE. OVER IN THE IBM WORLD WE HAVE TO DEPEND ON OUR SOFTWARE HAVING PRINTER DRIVERS OR WE CAN'T UTILIZE OUR NEW EQUIPMENT. LUCKILY WITH TI-WRITER'S TRANSLITERATE CODE CAPABILITY, WE CAN WRITE OUR OWN, AND THAT IS WHAT MARTY IS DOING. YOU PROBABLY KNOW THAT HE HAS AN IBM COMPATIBLE (WHICH I DON'T THINK HE EVER USES), AND HE DUMPS THE PRINTER DATA OUT IN ASCII SO THAT HE CAN USE IT ON THE TI.

HE IS STARTING TO DO SOME WILD AND CRAZY THINGS WITH THIS LASER, BUT DON'T LOOK FOR ANY TUTORIALS AS HE SAYS IT IS JUST TOO COMPLICATED. AT THE MOMENT I DON'T KNOW OF ANYONE WHO CAN DO GRAPHICS ON A LASER WITH THE TI, BUT PERHAPS THAT WILL COME ALSO. LET'S HOPE, AND SOME OF THE REST OF US MIGHT CONSIDER GOING THAT ROUTE!

CONTENTS

TI-CHIPS (21 JUL 1990)
DENNIS LIKENS

TI-CHIPS 12 JULY MEETING GOT UNDER WAY AT 1020 A.M. BY BEING CALLED TO ORDER BY PRES MATT ANDEL. MINUTES FROM JUNE'S MEETING WERE READ AND AMENDED . THE CHANGE WAS TO CLEAR UP ANY POSSIBLE MISUNDERSTANDING THAT OUR LIBRARY MANAGER, HARRY HOFFMAN PURCHASED THE TAPES OF THE LIMA CONFERENCE AND COPIES WERE BEING MADE BY JACK KORYTA. THANKS TO BOTH OF THESE GENTLEMEN .

LIN SHAW GAVE THE TREASURER'S REPORT. TI-CHIPS TOTAL IS $973.85. ALSO CAROLYN SHAW PURCHASED A SUITCASE FOR THE CLUB THAT IS TO BE USED TO CARRY THE PANASONIC PRINTER BOUGHT FOR THE CLUBS SYSTEM. THERE'S PLENTY OF ROOM FOR THE PRINTER AND A FEW OTHER ITEMS. THANKS CAROLYN. JOHN PARKEN REPORTS THAT TAPES AND MODULES STILL HOLDING STRONG. HE ALSO REPORTED THAT TEXAS INSTRUMENTS SENT HIM A LISTING OF CURRENT USER GROUPS/CLUBS FOR THE 99/4A. SEE HIM FOR DETAILS.

HARRY HOFFMAN REPORTS LOTS OF NEW PROGRAMS IN THE LIBRARY. ONE THAT WE ALL KNOW AS TIPS. HARRY HAS VERSION 1.6. HE SAYS BRING HIM YOUR VER 1.5 AND HE WILL COPY THE NEW VERSION OVER AT NO CHARGE. BE SURE TO GIVE YOUR OLD DISK. HE ALSO WANTED THE MEMBERS TO KNOW THAT IF THE WEATHER LOOKS BAD DURING THE DAY OF THE MEETING, HE WOULD NOT BE BRINGING THE LIBRARY OF DISKS TO THE MEETING. WOULDN'T WANT TO RUIN A FEW HUNDRED DISKS SO BETTER SAFE THAN SORRY.

MATT ASKED OF ANYONE HAD USED ANY OF THE LISTED "RECOVER FROM FCTN QUIT" AND GOT THEM TO WORK? THE ONE LAST PUBLISHED IN MICROPENDIUM DIDN'T WORK FOR HIM. VICE PRES GLENN BERNASEK INQUIRED ABOUT A PROGRAM THAT WILL LOAD FROM XB BUT NOT REQUIRE/USE THE 32K OF THE PEB. IN THE CHIPS LIBRARY IS A PROGRAM TO DO THIS, DISK 60/UTIL CALLED "MAKELOADER". THIS PRO WILL LIST THE PROS ON THE SCREEN AND YOU CAN SELECT/RUN THE PRO SELECTED. MATT DEMO'D THE MOUSE FOR THE 99. HE USED THE EXAMPLE PRO'S THAT CAME WITH THE MOUSE AND HE ALSO USED TI ARTIST+. THE MOUSE SURE MAKES THINGS A LOT EASIER/FASTER WHEN RUNNING ARTIST. THE MOUSE COMES WITH REQUIRED CABLES AND DOC'S. THE COST MAY VARY WITH VENDORS SO GIVE RON A CALL FROM RAMCHARGED COMPUTERS FOR INFO.

THE WIZARD OF XB, LES KEE, DEMO'D A PRO THAT HELPS OBTAIN THE STATIC BALANCE OF A DISC. THIS PRO WILL GIVE THE SPACING FOR WEIGHTS PLACED ON A DISC TO GET EVEN DISTRIBUTION. DESIGNING A SPINNING WHEEL? THEN CHECK OUT OUR THIS PROGRAM CAN HELP.

MATT DEMO'D SPELL IT! THIS PROGRAM FROM CHRIS BOBBITT IS THE SECOND OF TWO SPELL CHECKERS FOR THE 99. IT'S FAST, EASY TO USE AND A LITTLE MORE FLEXIBLE THAN THE ONE FROM DRAGON SLAYER. A NEW VERSION IS OUT SO BE SURE TO ASK FOR THE NEW VERSION AND REGISTER IT TO BE INFORMED OF NEWER VERSIONS. THE ONLY FUNCTION THAT STILL BOTH THESE PROGRAMS LACK IS A RETURN TO TIW OR FW AFTER YOUR FINISHED. THE OLE FCTN QUIT IS REQUIRED THEN YOU HAVE TO REBOOT YOUR WORDPRO.

WELL THAT ABOUT COVERS IT FOR THE MEETING. NEXT MEETING IS AUGUST 18. MATT ANDEL IS BRINGING HIS RAM CARD TO DEMO FOR THOSE LOOKING TO GETTING ULTIMATE SPEED AND NOT FORKING OVER BIG BUCKS FOR A HARD DRIVE AND CONTROLLER. THE CARD REQUIRES NO CONTROLLER CARD AND IS BATTERY BACKED. HOWIE WINKLER FROM NORTH COAST MAKES THESE CARDS FOR ALMOST HALF THEN LISTED PRICES SEEN IN PUBLICATIONS. IT WAS SAID THAT A 384K CARD GOES FOR ABOUT $160 AND A 1MEG $640. YOURS TRULY WON THE RAFFLE. CAN'T WAIT TO EXPLORE MY PLUNDER. TILL NEXT MEETING. ALOHA... SEE YA .

DENNIS
TICHIPS

NORTHCOAST NOTES
AUGUST, 1990

THE MEETING WAS CONDUCTED BY OUR CO-PRESIDENT BOB KAGY. WITH LAST-MINUTE VACATIONS AND NO NEWSLETTER TO REMIND US, WE STILL HAD ABOUT 25 IN ATTENDANCE.

ALMOST ANOTHER 40 DISKS WERE ADDDED TO THE LIBRARY AND 30+ WERE ADDED AT THE JULY MEETING. THESE ARE THE DISKS OBTAINED IN LIMA IN MAY. THERE WILL PROBABLY BE ANOTHER 30 OR SO AVAILABLE AT THE SEPTEMBER MEETING. IF YOU ARE A "GAMER" THERE ARE LOTS OF UNRELEASED TI-MODULES IN THIS GROUP. SOME REQUIRE A GRAM DEVICE (GRAM KRAKER, PGRAM OR GENEVE) TO RUN. HOWEVER, THERE ARE ALSO A LOT THAT WILL LOAD FROM EA5 OR XB LOADERS WHICH ARE INCLUDED ON ALL DISKS. MOST OF THE OTHER DISKS ARE GRAPHICS. THE LIST TAKES UP TOO MUCH ROOM TO PUBLISH IN THE NEWSLETTER, SO YOU WILL HAVE TO GET UPDATES FROM MARTY IF YOU WEREN'T AT THE MEETING.

WES RICHARDSON DEMO'D MAGNA-CALC, A PROGRAM WHICH HE WROTE TO ADD, SUBTRACT, MULTIPLY AND DIVIDE NUMBERS UP TO 240 DIGITS. IT IS AN XB PROGRAM WITH ASSEMBLY CALL LINKS TO SPEED UP THE CALCULATIONS, ESPECIALLY DIVISION. THIS PROGRAM WILL NOT WIN IN A SPEED DERBY, BUT AGAIN SHOWS WHAT SOMETHING WITH A LITTLE IMAGINATION CAN DO WITH OUR TI.

SEE YOU AT THE SEPTEMBER MEETING!

THIRD IN A SERIES OF FOUR.

COMPUTER COMMUNICATIONS BY BRUCE RODENKIRCH

THE REMARKABLE THING ABOUT MORSE'S DISCOVERY WAS THE DEVELOPMENT OF A CODE. ALTHOUGH MORSE'S CODE IS STILL USED IN PARTS OF THE WORLD THE INTERNATIONAL MORSE CODE, A REFINEMENT OF MORSE'S ORIGINAL CODE, IS MORE WIDELY USED. BUT THE IDEA OF A CODE REALLY OPENED THE DOOR FOR NEW AND EXCITING WAYS TO SEND INFORMATION OVER WIRES OR THROUGH USE OF RADIO SIGNALS. LET'S ANALYZE THE CODE. IT HAD THREE ELEMENTS, A DOT, A DASH AND A SPACE. TIMING IS ALSO A FACTOR. THE DASH IS THREE TIMES AS LONG AS THE DIT; THE SPACE BETWEEN CHARACTERS IS EQUAL IN LENGTH TO THE DOT AND THE SPACE BETWEEN WORDS IS EQUAL IN LENGTH TO THE DASH. THE LENGTH OF THE DOT IS DETERMINED BY THE SPEED OF TRANSMISSION AND THE REST ARE FIXED BY THEIR RELATIONSHIP TO THE DOT.

I SHOULD POINT OUT THAT MUCH OF THE CODE WHERE HUMANS ARE

INVOLVED MAY VARY IN THE LENGTH OF THE ELEMENTS, AND SOME OF THIS IS INTENTIONAL. FOR EXAMPLE, CODE CAN BE SENT AT A FIXED CHARACTER SPEED, SAY 10 WORDS PER MINUTE (WPM) WITH LONGER SPACES BETWEEN WORDS. THIS HELPS IN COMPREHENSION AND ALSO IN FUTURE INCREASES IN CODE SPEED DECIPHERING ABILITY. MOST AMATEUR RADIO OPERATORS OPERATE AT SPEEDS RANGING BETWEEN 5 AND 25 WPM ALTHOUGH SOME VERY SKILLED OPERATORS CAN HANDLE MUCH HIGHER SPEEDS. THIS IS STILL PRETTY SLOW COMPARED TO THE HUNDREDS AND EVEN THOUSANDS OF WPM TRANSMITTED BY COMPUTER.

THE MORSE CODE HAS ADVANTAGES FOR TRANSMITTING INFORMATION WHERE THE CODE IS GENERATED BY HUMAN OPERATORS BUT HAS SHORTCOMINGS FOR MACHINE GENERATION. IT DOES NOT HAVE UPPER AND LOWER CASE CHARACTERS AND IS LIMITED AS FAR AS THE NUMBER OF PUNCTUATION MARKS. THERE IS ALSO MORE CHANCE FOR ERROR.

ABOUT 1874, EMILE BAUDOT, (PRONOUNCE BAW DOUGH), A LIEUTENANT IN THE FRENCH TELEGRAPY SERVICE DEVELOPED THE BAUDOT DISTRIBUTOR FOR TELEGRAPHY. THE BAUDOT CODE WAS IMPROVED BY A MAN NAMED MURRAY AND THIS CODE WAS NAMED EITHER THE MURRAY CODE, OR THE MURRAY/BAUDOT CODE. UNLIKE THE MORSE CODE, WHICH USES DOTS AND DASHES IN GROUPS OF VARYING LENGTHS, THE BAUDOT CHARACTERS ARE MADE UP OF THE SAME NUMBER OF UNITS OR BITS. BAUDOT CHARACTERS ARE COMPOSED OF FIVE BITS OF ZEROS AND ONES WHICH LIMITS THE NUMBER OF CHARACTERS TO THIRTY TWO (2 TO THE FIFTH POWER). SINCE THE NUMBER ZERO IS NOT USUALLY USED THIS IS THIRTY ONE COMBINATIONS. THIS IS NOT ENOUGH TO SEND THE 26 LETTERS OF THE ALPHABET, 10 DIGITS AND THE NINE COMMON PUNCTUATION CHARACTERS, SO TWO OF THE CHARACTERS ARE DESIGNATED AS "SHIFT" CHARACTERS.

ONE OF THESE IS BYTE 11011, OR "FIGS", WHICH IS A LIST OF DIGITS, PUNCTUATION MARKS AND PRINTER COMMANDS. THERE ARE TWO LISTS OF FIGS, ONE IS FOR U.S. FIGURES AND ANOTHER ONE FOR CCITT NO. 2 FIGURES, WHICH STANDS FOR THE INTERNATIONAL SYMBOLS ESTABLISHED BY THE INTERNATIONAL CONSULTIVE COMMITTEE FOR TELEPHONE AND TELEGRAPH CCITT NO. 2 CODE ARRANGEMENT. WHEN DATA SET 11011 IS SENT THE RECEIVING MACHINE SWITCHES TO "FIGS", AND DECODES EACH FOLLOWING CHARACTER AS DESIGNATED IN THE FIGURE LIST, AND WHEN BYTE 11111, FOR LTRS, IS SENT THE PRINTER SWITCHES BACK TO THE LETTERS LIST. THE SYSTEM LACKS LOWER CASE LETTERS BUT THIS WAS NOT A SERIOUS DRAWBACK AT THE TIME IT WAS DEVELOPED.

ALTHOUGH FIVE BITS WERE USED TO DENOTE EACH CHARACTER, TWO OTHER BITS WERE NEEDED IN THIS SYSTEM. THESE WERE THE "START" AND "STOP" BITS, ANALOGOUS TO THE "MARK" AND "SPACE" BITS USED IN EARLY MECHANICAL TRANSMISSION OF TELEGRAPHY. SIMPLY PUT, THE RECEIVER HAS TO KNOW WHEN A CHARACTER BEGINS AND ENDS. IMAGINE A TELEGRAPH WIRE WITH A CONSTANT OR "RESTING" VOLTAGE, WAITING FOR A SIGNAL TO BE SENT. WHEN THE FIRST CHARACTER IS SENT THE VOLTAGE DROPS TO ZERO, WHICH IS THE "START" BIT. THE RECEIVER WAITS FOR A SPECIFIED PERIOD OF TIME (DEPENDING ON THE TRANSMISSION SPEED BUT THE SAME TIME REQUIRED FOR A DATA BIT) AND THEN DURING FIVE SUCCEEDING TIME INTERVALS RECORDS THE VOLTAGE STATE, ASSIGNING A ONE FOR A VOLTAGE JUMP AND A ZERO FOR NO CHANGE. THE VOLTAGE THEN GOES TO THE RESTING VOLTAGE FOR A TIME PERIOD LONGER THAN THAT FOR THE DATA BIT. THIS IS THE "STOP" BIT. THE SEQUENCE THEN REPEATS FOR THE NEXT LETTER. THE BITS ARE SENT WITH THE LEAST SIGNIFICANT BIT SENT FIRST, OR IN ORDER FROM RIGHT TO LEFT. FOR THE LETTER "D", 01101, THE FIRST BIT SENT AFTER THE START BIT IS A ONE.

THERE IS ONE INTERESTING SIDELIGHT TO THE BAUDOT CODE. SINCE MACHINE TRANSMISSION OF THIS CODE WAS DONE WITH PUNCHED TAPES, THE ASSIGNMENT OF BINARY CODES TO THE LETTERS WAS DONE SO THAT THE FREQUENTLY USED LETTERS USED FEWER "ONE'S" WHICH MEANT LESS PUNCHING. FOR EXAMPLE, THE LETTER "E" IN BAUDOT IS 00001 AND "X" IS 11101.

IN 1968 THE AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI) ADOPTED THE AMERICAN NATIONAL STANDARD CODE FOR INFORMATION INTERCHANGE (ASCII). THIS CODE USES SEVEN DATA BITS PLUS A EIGHTH BIT, OR PARITY BIT FOR ERROR CHECKING. THE SEVEN DATA BITS ALLOWS 128 CHARACTERS (TWO TO THE SEVENTH POWER-ASCII ZERO IS NULL). YOU ALL HAVE LISTS OF THE ASCII NUMBERS AND THE CHARACTERS THEY REPRESENT. THESE BYTES ARE PRECEDED BY A START PULSE AND ENDED WITH A LONGER STOP PULSE, AS IN THE MURRAY/BAUDOT CODE.

THE PARITY CODE WORKS AS FOLLOWS. PARITY CAN BE PREDETERMINED TO BE ALWAYS MARK (1), ALWAYS SPACE (0), EVEN PARITY, OR ODD PARITY. IF THE PROTOCOL IS SET AT ODD PARITY, THE TOTAL NUMBER OF BITS IN THE BYTE WILL BE ADJUSTED TO AN ODD NUMBER BY THE EIGHTH BIT. FOR EXAMPLE, IF THE NUMBER OF ONES IN THE SEVEN DATA BITS IS EVEN, THE EIGHTH BIT IN THE BYTE IS A ONE. FOR MOST OPERATIONS THIS IS NOT TOO IMPORTANT BECAUSE THE SYSTEMS WE USE DO NOTHING ABOUT AN ERROR ANYWAY. IN ADDITION THE ERROR MAY BE CONCEALED IF TWO DATA BITS ARE WRONG. THUS, NO ERROR CORRECTION IS DONE EXCEPT IN MORE SOPHISTICATED SYSTEMS. IF NO PARITY IS SPECIFIED THE EIGHTH BIT IS USUALLY SET TO A SPACE OR ZERO. FOR CLARIFICATION, THE EIGHTH BIT IS THE MOST SIGNIFICANT BIT (MSB) OR THE ONE TO THE LEFT.

MY AMATEUR RADIO CALL IS KX8U AND I HAVE BEEN PUTTING MY TI99/4A TO GOOD USE ON THE HAM BANDS. WE USE RADIO TELETYPE (RTTY)USING BAUDOT CODE, RADIO TELETYPE (ASCII), AMTOR ERROR CORRECTING RTTY, PACKET ERROR CORRECTING RTTY AND INTERNATIONAL MORSE CODE. AS YOU CAN SEE THE BASIC DIFFERENCES ARE IN THE CODE USED, BAUDOT OR ASCII, AND IN ERROR CORRECTION.

THIS SEEMS LIKE A GOOD PLACE TO STOP. NEXT TIME I WILL DESCRIBE HOW THIS TYPE OF DATA TRANSMISSION IS USED IN AMATEUR RADIO. THIS STARTED OUT AS A TWO PART SERIES AND NEXT TIME WILL BE THE FOURTH AND LAST, I HOPE.

SEE YOU NEXT MONTH.

MAGNA-CALC

by WESLEY R. RICHARDSON

NORTHCOAST 99ERS, CLEVELAND, OH

MAGNA-CALC is a program which can add, subtract, multiply and divide numbers up to 240 digits long, with exponents from -99,999 to +99,999. The X and Y registers are used for calculations, and there are 10 additional storage registers. Why would anyone have a need for this number of digits? I don't know, but if you do, this program is for you.

OPTIONS

The main menu options are +-*/ Copy, Edit, New, Print, Quit, Recall, Save, View, Write, and eXecute. Each of these options is invoked by pressing the math symbol or the capital letter of the word. Copy will copy from any register to any other register. Edit allows modification of a previously entered number. New clears a register to zeros and allows input of a new number. Print prints a listing of all 12 registers to the printer, and uses one full page. Quit is to leave the program. Recall allows the registers 0 through 9 to be recalled from disk. Save stores the values in registers 0 to 9 on disk in DIS/VAR 80 format. View allows viewing of any register, but does not allow modificaton of that register. Write provides a means to program up to 252 steps of main menu key presses. Any of the functions or 12 registers can be used. The eXecute option will then run from 0 to 99999 repetitions of the the program steps from the Write option.

A color block on the screen is green when you can input with a single key press. When it is blue, the enter key is required after your input. White indicates calculations or input/output. Red is when the program is between routines. During white or red, the program will not accept input.

Addition or subtraction of two 240 digit numbers takes approximately 20 seconds, while multiplication takes just over 5 minutes. Division of 240 digit numbers takes an average of 8 minutes, with a maximum of 11 minutes. The division routines are executed almost entirely in Assembly, but are documented as REM statements for the equivalent operations in XBASIC. All of the routines have shorter times when trailing zeros are found.

One caution with MAGNA-CALC is that no rounding is done, but rather the answer in the X register is truncated at the 240th digit. Multiplication is done with 480 digit precision, but is also truncated at the 240th digit.

The numbers in MAGNA-CALC are represented as floating point numbers in the form: +1.2345 E+67890 which is the same that BASIC uses for large or small numbers. The "E" simply says to raise 10 to the power of the second group of numbers. For the example above it would be the number 1.2345 with the decimal point shifted 67,890 positions to the right. If the exponent is negative, then shift the decimal that many places to the left. Other examples are:

```
+9.8765 E+00003 is equal to  9876.5
-9.8765 E+00003 is equal to -9876.5
+9.8765 E-00004 is equal to     0.00098765
-9.8765 E-00004 is equal to    -0.00098765
```

The Extended BASIC manual gives further examples of this scientific notation for numbers.

The files associated with this program are:

```
MAGNA-CALC XBASIC program, LOAD, then RUN
MAGNA-DATA DIS/VAR 80 data file of constants
MAGNA-DOC  this documentation
MAGNA-ED/S Assembly source code for divide
MAGNA-ED/O Assembly object code for divide
MAGNA-PI   DIS/VAR 80 factors using Pi
```

To use the program, go into Extended BASIC and type OLD DSK1.MAGNA-CALC. Because of the size of this program, it is stored in INT/VAR 254 format, and cannot be run directly from a LOAD program. After the program has loaded, type RUN. The program will load the MAGNA-ED/O file from disk 1.

The MAGNA-DATA file contains the square roots of 2 and 3, the natural log base e, pi, the speed of light c, the charge on an electron e-, Avogadro's number N, Planck's constant h, and pi squared.

The MAGNA-PI file contains the following all to 240 digits: Pi/180, 1/Pi, SQR(Pi), Pi, 4*Pi/3, 1/SQR(Pi), 2*Pi, Pi*Pi, 4*Pi*Pi, and 180/Pi, where Pi=3.14159...

Y TO A POWER

The Write and eXecute functions are a way for you to program MAGNA-CALC to do repetitive operations without your intervention. Integer powers of numbers can be found by Writing the formula *VXV. Note when using V in a formula, be sure to give the next character as a valid register, and then any other valid character to simulate the "press any key" to return from View.

Start with the value of +1.00000 in register (1) and the base number in (Y). Then eXecute the program the number of times equal to the power. If you want 2 to the 5th power, start with +2.0000 in (Y) and eXecute the above formula 5 times. The answer is +3.2000 E+00001 or 32.

FACTORIAL

To calculate n factorial, for example, which is (n) * (n-1) * (n-2) ... * 1. Store +1.0 in register (1), +0.0 in (2), and +1.0 in (3). Then Write: C1YC2X+CX2C3Y*CX3. When you eXecute this program n times, n factorial will be in register (3).

TRIANGLE AREA

If you wish to find the area of a triangle using A=(1/2)*b*h where b=base and h=height, then start with register (1) as +5.0000 E-00001 (or 1/2), the value for b in register (2), and h in (3). Write the formula C1XC2Y*C3Y*CX4, and eXecute it once. The answer for the area A will be in (4).

VALUE OF E

With the programming ability, MAGNA-CALC can be used to calculate numbers generated from a series, like the natural log base e, for example. Store +1.0 in register (0), and +0.0 in (2). Copy (0) to (1), and Copy (0) to (3). Write the program: C1YC2X+CX2C3Y*CX3C1Y/C0Y+CX0. Then eXecute it 150 times (about 11 hours) and you will have calculated e to close to 240 digits.

SQUARE ROOTS

You can use MAGNA-CALC to calculate square roots of positive numbers, if you follow the steps below. Register 1 is indicated as (1). This program can take up to 3 hours for a 240 digit number, but will run automatically after the first few steps. Let the starting number be +1.2345 67890 E+33333 for example.

1. Write down the first 10 digits and the exponent of the number you want a square root of. 2. From the main menu of MAGNA-CALC, press FCTN 4 to break the program. 3. Use steps 4 and 5 (odd exponent) or steps 6 and 7 (even exponent) to estimate the square root of your number.

4. If the exponent is odd, subtract 1, then divide by 2, thus an exponent of +33333 becomes +16666, while -33333 would be -16667. 5. If the exponent was odd, type PRINT SQR(12.34567890) with two digits to the left of the decimal point. The answer is 3.513641829, go to step 8.

6. If the exponent is even, just divide by 2, thus +33332 would by +16666, and -33332 would be -16666. 7. If the exponent was even, type PRINT SQR(1.234567890) with one digit to the left of the decimal point. The answer is 1.111111106.

8. Write down the estimates for the square root and it's exponent from steps 4+5 or 6+7. 9. Type CON to continue. 10. Press V, then X, then any key.

11. Put the number you want the square root of, like 1.2345 67890 E+33333 in (1). 12. Using New, put the estimated value for the square root, and exponent in (2), keeping the sign for the exponent from step 4 or 6.
13. Using New, put +1.0000 E+00000 in (3).
14. Using New, put +2.0000 E+00000 in (4).
15. Write the following program steps, then eXecute it 1 time: C1YC4X*C3Y/CX4C2Y*CX3
16. Write the following program steps, then execute it 6 times: C2YCYX*CXYC1X-CX0C3Y*C2Y-CX2 C4Y*CX3.
17. The square root of (1) will be found in (2), and the error will be in (0). When the error reaches zero, or a number which is E-00239 less than the exponent in (1), then further iterations will probably not reduce the error. (3) and (4) are factors used for adjusting the square root estimate.

SINE OF X

The calculation of trigometric functions takes several hours because the series converges slowly, but the SIN(X) can be calculated using the following steps. 1. Start with X degrees in the (X) register. 2. Recall MAGNA-PI, which will have Pi/180 in register (0). 3. New +1.0 in (1) and New +0.0 in (4).

4. Write C12C13C45C46C47C48C49CX8C0Y* CX9CX7CX6CXY*C5Y-CX4 and eXecute it 1 time.
5. Write C6YC4X*CX6C2YC1X+CXYC1X+CX2* C6Y/CX6C7Y+CX7V7V and eXecute it 80 times, about 14 hours.
6. X in degrees will be in (8), in radians in (9) and SIN(X) will be in (X) and (7). The last increment to SIN(X) will be in (6).

900811WR

Filename: MAGNA-DATA MAGNA-CALC by WR

```
REGISTER 0   SQR(2)
 +1.41421356237309504880168872420969807856967187537694807317667
   97379907324784621070388503875343276415727350138462309122970 2
   49248360558507372126441214970999358314132226659275055927557 9
   99505011527820605714701095599716059702745345968620147285174 2E+00000
REGISTER 1   SQR(3)
 +1.73205080756887729352744634150587236694280525381038062805580
   69794519330169088000370811461867572485756756261414154067030 2
   99699450949989524788116555120943736485280932319023055820679 7
   48201010846749232650153123432669033228866506722546689218379 7E+00000
REGISTER 2   e
 +2.71828182845904523536028747135266249775724709369995957496696
   76277240766303535475945713821785251664274274663919320030599 2
   18174135966290435729003342952605956307381323286279434907632 3
   38298807531952510190115738341879307021540891499348841675086 2E+00000
REGISTER 3   Pi
 +3.14159265358979323846264338327950288419716939937510582097494
   45923078164062862089986280348253421170679821480865132823066 4
   70938446095505822317253594081284811174502841027019385211055 5
   96446229489549303819644288109756659334461284756482337867831 6E+00000
REGISTER 4   c
 +2.99792458000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000E+00008
REGISTER 5   e-
 +1.60277330000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000E-00019
REGISTER 6   N
 +6.02213670000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000E+00023
REGISTER 7   h
 +6.62607550000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000E-00034
REGISTER 8   Pi*Pi
 +9.86960440108935861883449099987615113531369940724079062641334
   93762200448224192052430017734037185522318240259137740231440 7
   77723481220300467276106176779851976609903998562065756305715 0
   60412328403287808693527693421649396665715190445387352617793 8E+00000
REGISTER 9   Big
 +9.99999999999999999999999999999999999999999999999999999999999
   99999999999999999999999999999999999999999999999999999999999
   99999999999999999999999999999999999999999999999999999999999
   99999999999999999999999999999999999999999999999999999999E+99999
                                                    900811WR
```

Filename: MAGNA-PI MAGNA-CALC by WR

REGISTER 0 Pi/180
+1.74532925199432957692369076848861271344287188854172545609719
144017100911460344944368224156963450948221230449250737905924
838546922752810123984742189340471173191682450150107695616975
535812386053051687886912711720870329635896026424901877043508E-00002
REGISTER 1 1/Pi
+3.18309886183790671537767526745028724068919291480912897495334
688117793595268453070180227605532506171912145685453515916073
785823692229157305755934821463399678458479933874818155146155
492793850615377434785792434795323386724780483447258023664760E-00001
REGISTER 2 SQR(Pi)
+1.77245385090551602729816748334114518279754945612238712821380
778985291128459103218137495065673854466541622682362428257066
623615286572442260252509370960278706846203769865310512284992
517302895082622893209537926796280017463901535147972051670019E+00000
REGISTER 3 Pi
+3.14159265358979323846264338327950288419716939937510582097494
459230781640628620899862803482534211706798214808651328230664
709384460955058223172535940812848111745028410270193852110555
964462294895493038196442881097566593344612847564823378678316E+00000
REGISTER 4 4*Pi/3
+4.18879020478639098461685784437267051226289253250014109463325
945641042187504827866483737976712282275730953078201770974219
612512614606744297563812544171308156600378803602584694807741
285949726527324050928590508130088791126150463419764504904420E+00000
REGISTER 5 1/SQR(Pi)
+5.64189583547756286948079451560772585844050629328998856844085
721710642468441493414486743660202107363443028347906361707351
689931494826162866365489520017768993292837637059598439760352
464350217972571211580245772820220554508527173216622208463300E-00001
REGISTER 6 2*Pi
+6.28318530717958647692528676655900576839433879875021164194988
918461563281257241799725606965068423413596429617302656461329
418768921910116446345071881625696223490056820540387704221111
928924589790986076392885762195133186689225695129646757356632E+00000
REGISTER 7 Pi*Pi
+9.86960440108935861883449099987615113531369940724079062641334
937622004482241920524300177340371855223182402591377402314407
777234812203004672761061767798519766099039985620657563057150
604123284032878086935276934216493966657151904453873526177938E+00000
REGISTER 8 4*Pi*Pi
+3.94784176043574344753379639995046045412547976289631625056533
975048801792896768209720070936148742089272961036550960925763
110893924881201869104424707119407906439615994248263025222860
241649313613151234774110773686597586662860761781549410471175E+00001
REGISTER 9 180/Pi
+5.72957795130823208767981548141051703324054724665643215491602
438612028471483215526324409689958511109441862233816328648932
814482646012483150360682678634119421225263880974672679263079
887028931107679382614426382631582096104604870205064442596560E+00001

900811WR

TI-Base -By- Inscebot, Inc.

P.O.Box 291610, Pt.Orange, FL 32129

Version 3.01 Tutorial 21.1.1 By *Martin A. Smoley*
NorthCoast 99'ers User Group - July 21, 1990

Inventory Control

Inventory Control is a fancy name for reordering parts when your stock gets low. In the June issue we created five databases and filled them with part numbers, prices, etc. In those DBs we entered Current Stock (CRS), Minimum Stock (MNS) and Maximum Stock (MXS) to use for stock control. This month I have written two small CFs to check those fields and copy certain items to an ORDER Db if they meet my requirements. To put it very simply, if the Current stock falls below the Minimum stock, reorder. My first step is to CLOSE ALL of the currently open Dbs. My second step is to utilize that wonderful INSTALL area again. For those of you who do not have ramdisks, the INSTALL area is a great new feature. It's fast, quiet and doesn't wear out your disk drive. INSTALL ADD DSK2.\ORD places or loads the entire CF named \ORD into the TI's VDP memory. Aside from the previously mentioned advantages I also wanted to demonstrate INSTALLs ability to perform intricate steps, such as Math or WHILE loops. When you get the hang of it I'm sure you will use INSTALL quite frequently. SELECT 2 and USE DSK2.ORDER merely opens the ORDER Db in slot 2. You should recognise the lines from here to ENDCASE, they are straight out of LSPRNT/C from last month. I merely edited LSPRNT by deleting and adding lines and saved it to the new name ORDPRNT/C. As in LSPRNT the DOCASE is used to open each of the DBs. The WHILE .NOT. (EOF) will leaf through each DB, one record at a time. IF the Current Stock (CRS) is less then the Minimum Stock (MNS), \ORD will be executed. IF not then the CF will MOVE to the next record and try again. If you look at \ORD, you can see that most of its line merely moves data from the 74LS Db to the ORDER Db, after a new record has been APPENDed. When using (or SELECTing) slots in this manner always tell TIB where the data is located by slot number (2.COPNM, 1.COPNM etc.). There are two field changes from the 74LS to ORDER, one is ORDQT (for ORDer QuanTity) and CHK. ORDQT should be self explanitory. CHK will be a number half way between MNS and MXS. If there are no other determining factors, I would like to raise my stock level to the first whole number above CHK. I saved CHK in the new Db (which is not necessary) to allow myself to visually check the process. The sum of CRS and ORDQT should be slightly larger then CHK.

```
*            07/08/90        ORDPRNT/C
CLOSE ALL
 INSTALL ADD DSK2.\ORD
   SELECT 2
   USE DSK2.ORDER
   SELECT 1
 LOCAL LOOP N 3
 REPLACE LOOP WITH 1
WHILE LOOP<6
  DOCASE
   CASE LOOP = 1
    USE DSK2.74LS'S1
   BREAK
   CASE LOOP = 2
    USE DSK2.74LS'S2
   BREAK
   CASE LOOP = 3
    USE DSK2.74LS'S3
   BREAK
   CASE LOOP = 4
    USE DSK2.74LS'S4
   BREAK
   CASE LOOP = 5
    USE DSK2.74LS'S5
   BREAK
  ENDCASE
    WHILE .NOT. (EOF)
     IF (1.CRS)<(1.MNS)
      DO \ORD
     ENDIF
      SELECT 1
      MOVE
    ENDWHILE
   CLOSE
  REPLACE LOOP WITH LOOP + 1
ENDWHILE
  CLOSE ALL
 INSTALL REMOVE \ORD
RETURN  © Martin A. Smoley 1990
```

```
*            07/08/90       \ORD/C
 SELECT 2
 APPEND BLANK
REPLACE 2.COPNM WITH 1.COPNM
REPLACE 2.MFGPARTNUM WITH 1.MFGPARTNUM
REPLACE 2.CPRICE WITH 1.CPRICE
REPLACE 2.CRS WITH 1.CRS
REPLACE 2.ORDQT WITH (1.MXS-1.MNS)
REPLACE 2.CHK WITH ((1.MXS - 1.MNS) /2);
 + 1.MNS
 WHILE (2.ORDQT+2.CRS)<2.CHK
  REPLACE 2.ORDQT WITH 2.ORDQT+1
 ENDWHILE
REPLACE 2.LCTN WITH 1.LCTN
REPLACE 2.LASTSALE WITH 1.LASTSALE
REPLACE 2.LRESTOCK WITH 1.LRESTOCK
REPLACE 2.NSN WITH 1.NSN
REPLACE 2.DESC WITH 1.DESC
RETURN  © Martin A. Smoley 1990
```

Continued Next Page.

Version 3.01 Tutorial 21.1.1 By *Martin A. Smoley*
NorthCoast 99`ers User Group - July 21, 1990

This process starts with REPLACE 2.CHK WITH (1.MXS - 1.MNS). If the maximum is 10 and the minimum is 5 and we know that the current stock is below 5 then we can safely order 5 items without going over the maximum. For my own preferance, I'd like to see the stock level greater that the half way point between MNS and MXS. By mental deductions I can see that half way between 5 and 10 would be 7.5. REPLACE 2.CHK WITH ((1.MXS - 1.MNS) /2) + 1.MNS does my mental calculation and places the answer in CHK. As you can see, this process will not produce a whole number, so I have set CHK to one decimal place. If you set a field to zero decimal places the field will be truncated to a whole number value. The WHILE loop checks to see if the amount we have in stock plus the amount we should order are less then the number we are CHecKing. If so, it adds 1 to the ORDQT until (ORDQT + CRS) is no longer less then CHK. This way of establishing our new stock level is slow, cumbersome and poor programming technique, but I do not consider those thoughts. This sequence will work, it will produce the end result we want and (most important) it will give the non programmer an opportunity to understand the thought process that he or she must use and the calculating process that TIB must use to find the desired answer. You may also want the ordering CF to scrutanize the prospective fields in greater detail. For example, you may want TIB to look at the last sale date and compare it to the last restocking date. If a large amount of time has elapsed you may ~~not~~ want to restock as many items. If you sell out in a short period of time, you may want to double or at least increase your order. I also decided to direct my order list to another Database, rather than the printer, as the CFs name might indicate. This is because the price and availability may have changed after your last restock. If you use a Database, as I have, you can look at the data and decide if you want to order more or less of a particular item, based on its popularity. You can also check the current prices against a catalog or by other means. After editing the ORDER Db to your satisfaction, another CF can be used to write out an order form for the materials you require. If you have a normal supplier list, which you would keep in a normal supplier Database, sorted by their Normal Supplier Number (NSN), TI-Base can break down the ORDER Db and send orders to each supplier for those parts marked with that specific number. This seems like a lot of work if you consider my example (IC chips costing from 14 to 50 cents each), but the same ideas can be applied to larger items costing much more. And, if any of you are regular shoppers at Radio Shack you know that even for a 69 cent purchase, they run the barcode reader over the package, they ask for the last four digits of your phone number and your last name and their computer does the rest. As a matter of fact, the smaller your profit margin is the more advantagious the computer inventory control becomes.

Ordering Updates

I have decided to not be involved in the distrabution of the new TI-Base updates. If you received any updates from me in the past, I suggest you contact Dennis Faherty by mail for your future updates. Dennis is cheerful, courtious and helpful, and except for the fact that you will probably have to send back your current diskettes to get the new update price, you should have no trouble dealing directly with Inscebot. I will start any future tutorials with Inscebot's address for your convenience. I think that the new updates will cost $14.95, but I'm not sure.

My Last Tutorial!

This tutorial and any future tutorials should each be considered my last tutorial. Because of many other demands on my time, I find it almost impossible to allocate the time needed to write the TI-Base Tutorials. I have not lost the interest, but I have lost the energy needed to get the job done. Therefore, this tutorial should be considered my last. If by some chance I find the time and mental ability to write another tutorial, then that should be considered my last. "You will no longer see (Continued Next Month.) at the end of the tutorials." No matter what happens I plan on throwing in the towel by the end of this year. It's been a lot of fun, but there are many things I would like to try when and if I find a little spare time.

Good luck. Marty.

****

CREATED 07/08/90 CHANGED 06/17/90

FIELD	DESCRIPTOR	TYPE	WIDTH	DEC
1	COPNM	N	005	00
2	MFGPARTNUM	C	010	
3	CPRICE	N	006	02
4	CRS	N	003	00
5	ORDQT	N	003	00
6	CHK	N	004	01
7	LCTN	C	004	
8	LASTSALE	D	008	
9	LRESTOCK	D	008	
10	NSN	C	003	
11	DESC	C	040	

ORDER Database Listing with STRUCTURE above.

REC	COPNM	MFGPARTNUM	CPRICE	CRS	ORDQT	CHK	LCTN	LASTSALE	LRESTOCK	NSN	DESC
0000	1005	74LS05	0.14	3	5	7.5	D2B1	04/15/90	08/21/89	24	Hex Inverter (Open Collector)
0001	1008	74LS08	0.14	2	6	7.5	D2B4	06/19/90	05/15/90	9	Quad 2-in AND Gate
0002	1038	74LS38	0.24	4	5	7.5	D5B3	02/25/90	01/20/90	24	Quad 2-in NAND Buffer (Open Collector)
0003	1373	74LS373	0.50	3	5	7.5	D7B2	05/02/90	01/09/90	24	Tri-State Octal Dual Latch

```
100 CALL CLEAR :: PRINT "WE
WILL USE JOYSTICK NUMBER TWO
" :: CALL SAY("WE WILL USE
JOYSTICK NUMBER TWO"):: PRIN
T :
110 PRINT "PRESS ANY KEY TO
START" :: CALL SAY("PRESS AN
Y KEY TO START")
130 CALL KEY(0,6,S):: IF S=0
 THEN 130
500 CALL COLOR(9,6,16):: CAL
L COLOR(10,6,16)
510 CALL CHAR(96,"1830707FCFF
FFFFF")
520 CALL CHAR(97,"0C183C7EFF
FFFFFF")
530 CALL CHAR(98,"060C1E3FFF
FFFFFF")
540 CALL CHAR(99,"03060F9FFF
FFFFFF")
550 CALL CHAR(100,"818387CFF
FFFFFFF")
560 CALL CHAR(101,"C0C1C3E7F
FFFFFFF")
570 CALL CHAR(102,"60C0E1F3F
FFFFFFF")
580 CALL CHAR(103,"3060F0F9F
FFFFFFF")
640 PRINT "HOW HOT DO YOU WA
NT YOUR    SOUP TO BE?":"IF
IT IS MORE THAN 100      DEG
REES IT IS TOO HOT!":
650 INPUT " THE TEMPERATURE
IS ":TEMP
660 SPEED=INT(TEMP/2):: IF S
PEED<50 THEN 670 ELSE PRINT
"TOO HOT!!" :: GOTO 650
670 PRINT "O.K."
1000 CALL CLEAR :: CALL MAGN
IFY(2):: CALL CHAR(136,"FF81
8181818181FF"):: RANDOMIZE
1110 R=4 :: TOL=5
1120 CALL SCREEN(16):: CALL
COLOR(1,6,6):: TARGET=1 :: C
ALL HCHAR(1,1,32,768):: H1=9
6
1130 FOR CHAR=65 TO 90
1140 SPRT=CHAR-64
1150 HV=INT(RND*40)-20 :: VV
=INT(RND*40)-20
1160 COLOR=INT(RND*12)+3 ::
IF COLOR=6 THEN 1160
1170 CALL SPRITE(#SPRT,CHAR,
COLOR,100,100,HV,VV)
1175 CALL HCHAR(1,1,H1,32):: 
H1=H1+1 :: IF H1=104 THEN H
1=96
1180 NEXT CHAR
1181 FOR SP=1 TO 24
1182 HHV=INT((RND*SPEED)-(SP
EED/2)):: VVV=INT((RND*SPEED
)-(SPEED/2))
1183 IF HHV=0 THEN 1182
1184 IF VVV=0 THEN 1182
1185 CALL MOTION(#SP,HHV,VVV
)
1187 NEXT SP
1190 CALL SPRITE(#27,136,2,1
00,100,0,0)
1200 CALL JOYST(2,X,Y):: CAL
L MOTION(#27,(-Y*8),(X*8)):: 
CALL KEY(2,GRAB,S)
1210 CALL HCHAR(1,1,H1,32):: 
H1=H1+1 :: IF H1=104 THEN H
1=96
1220 CALL COINC(#27,#TARGET,
TOL,I)
1230 IF I=0 THEN 1200
1240 CALL DELSPRITE(#TARGET)
:: PRINT CHR$(TARGET+64);
1250 TARGET=TARGET+1 :: IF T
ARGET<27 THEN 1200
1255 CALL DELSPRITE(ALL)
1260 PRINT : : : :: PRINT "D
O YOU WANT TO PLAY AGAIN?  I
F YES, PRESS Y, IF NO, PRESS
 N"
1270 CALL SAY("DO YOU WANT T
O PLAY AGAIN   IF YES, PRESS
 Y, IF NO PRESS N")
1280 INPUT "Y OR N":X$
1290 IF X$="Y" THEN 120
1300 IF X$="N" THEN 1320
1310 GO TO 1280
1320 CALL SAY("WE MUST PLAY
AGAIN SOME TIME.  GOODBYE")
```

CLEVELAND AREA 99/4A USERS GROUPS
C/O DEANNA SHERIDAN
20311 LAKE ROAD
ROCKY RIVER, OH 44116

CHECK YOUR EXPIRATION DATE.
THIS MAY BE YOUR LAST ISSUE!

FIRST CLASS RST CLASS

Exp Date: 91/07

FIRST CLASS IRST CLASS