<pre>1</pre>							
EAR 99'ERS							
⊑ast Anglia Region 99'ere Heer's Group							
VOLUME 2 - ISSUE 2 - JUNE '88							
Great Britain: United States:							
Scott Copeland SSgt D. S. Copeland EAR 99'ers User's Group 13 Elm Walk 1979C5/LGG Lakenheath, Suffolk FCS Box 5927 England IP27 90R APO NY 09179-5379 Tel: 063881-3457 Tel: 011-44-63881-3457							
President † Scott Copeland † 063881-3457 Vice-President † Robert Wordsworth † 0603-38832 Secretary/Treasurer † Jo Ann Copeland † 063881-3457 Publication's Officer † Bryan Cloud † 0473-464996							
t + + + + + + + + + + + + + + + + + + +							
tt Fg 2 tt Bytes and Pieces tt Fg 4 tt On the Exchange Front tt Fg 6 tt For Sale tt Fg 8 tt Adventuremania Adamitis tt Fg 10 tt In Appreciation tt Fg 11 tt Computer knowledge tt Fg 14 tt Mini-Memory Fart IX tby Robert Wordswortht tt Fg 20 tt For Big & Little Rids t Stock Market t tt Fg 24 tt FEB Speech Interface tt Fg 25 tt TI-99/4A Memory Architecture tt Fg 26 tt Cassette Corner t Gone Fishing! t tt Fg 28 tt Treasury Report & Next Meeting							
This newsletter is not copyrighted and may be reprinted in whole or in part. FLEASE give credit where credit is due, and notate the Giun and the Author. Any user assumes sole liability for use and please note any information by authors' does not guarantee it to be error-free.							
Now, turn the page for an EAR-ful of information							

BYTES AND PIFCES

First of all, we'd like to extend a hearty welcome to RONALD L. JOHNSON, who just joined our TI Family, along with TRISTAN DONDVAN who also became a member of our TI Family. Welcome aboard Ronald and Tristan! Ronald Johnson is trying to build up his system (currently on cassette) so if you decide to sell something or have something you no longer need, why not keep Ronald in mind? We also welcome PAUL HOEHN who also include our TI Family (and I think I forcet to show this in second also joined our TI Family (and I think I forgot to show this in recent issues!). Oh the shame of being a forgetful editor... (actually, an overworked one...) Gotta' have some kind of excuse, huh?

Secondly, we received our copy of LEGENDS Version 1.1 rather quickly from ASGARD SOFTWARE (great service') and if you'd like to see the newer version ask Scott. We were going to provide a list of the Monsters, along with their hit class and experience points, etc., but Scott found (after compiling the original list) that this has ALL changed so he'll have to play it a bit more before we provide the new list. Just can't wait huh?...

For those of you who receive PETER BROOKS' International TI-Lines Newsletter: Just a note to say that he's running a bit behind schedule newsietter: oust a note to say that he standing problems (actually, his due to an increased workload and transportation problems (actually, his year is going worse than ours!). Please bear with him and you'll see the year is going worse then ours: , Freese been with him and you is see the next issue in a bit. And Yes, he is alive and well (or at least alive!) and it was great speaking with him the other night. Hang in there here! (Update May 23) - Feter has called and advised the good news of a new job which he is excited about starting and things are finally looking (way) up for him! Congratulations Peter! We knew there wasn't any other way to go but up!

I NEED ARTICLES!

If there's ANYTHING you've been interested in, or working on, or discovered, or WHATEVER, please feel free to submit an article! Flease, the editor is begying (disquisting, isn't it?)...

The SIG meeting scheduled last month was a good one, although attendance wasn't what we expected it to be. However, it was informative and the smaller group that attended had the chance to ask those questions that were driving them crazy and the "on-hands" demonstration helped with a high the country thanks to Trany township to taking the time out to quite a bit! Our many thanks to TONY BOWDEN for taking the time out to attend the SIG and show us his expertise! More members are now on their way to putting those modules on disk in less than 5 minutes time!

PHILIP TROTTER mentioned I should remind everyone about the Binders you can obtain for the new format newsletter. Ever forgetful Jo is glad our members keep her on her took and here's the address and price for obtaining the Binders to hold your newsletters:

BINDERS	CORDEX Navy Blue Binding	2.95 inc. VAI
78 Wielley Road Wilpshire	9-75 X & WIREX Black Binding	2.95 inc. VAT
BLACKBURN Lancs BB1 9LF	8.75 X 6 Postage: 1 binder = 1.50 (extra) 4/6 = 3.00	2/3 = 2.00 $6+ = 3.50$

(This information gratiously supplied by both International TI-Lines Newsletter and TI*MES Newsletter.)

Scott is working on getting our BBS up and running so those of you with a modem better get ready! Details either in this issue or the next one! The last Scott said was "it would be ready anytime now"!

> East Anglia Region 97'ers User's Group Page 2

The HONESTY AWARD OF THE MONTH goes to both EDDY CARTER and PHILIP TROTTER for helping the Secretary/Treasurer keep her records accurate! Two slip-ups on renewal dates have been corrected due to our honest TI members! THANKS GUYS!

IN THIS ISSUE ...

In this issue: You'll find the usual Adventure Help Column, along with Robert's Mini-Memory Tutorial (THANE'S ROBERT!), plus the Big & Little Kids column. This month we add some items borrowed from WEST PENN 99'ERS — a PEB Speech Interface and Tl-99'/4A Memory Architecture 'map'. Also, here's your chance to test your computer knowledge in "45 Things You Always Wanted To Know About Computers But Were Afraid To Ask!' quiz. JoAnn had fun putting this together and surprisingly got 100' (maybe because she had the answers in the book?).

NOTES ON THE BIRTHDAY PARTY! 22 May 1988 by: JoAnn Copeland, Sec/Treas

We couldn't have asked for better weather (for a change!) the day of the party! The sun was shining (what an unfamiliar but welcome sight) and the back-yard was filled with adults and children alike!

I'd like to thank everyone now who brought such delicious desserts with them! Yes, I'm working off the added 10 pounds now and maybe by the next Bar-B-Que my weight will be just under that of a full-grown female whale! We have alot of talented Chefs in our group! My thanks to Scott Copeland and Jne Hunt for taking over the outdoor-chef duties and providing us with such well-cooked Chicken, Hot Dogs and Hamburgers! Everyone had their fill but we still have enough left over for another Bar-B-Que so Neil suggested we send everyone a free hamburger with their next newsletter!

We had approximately 30 attend and the guys finally broke down after eating and got a good of (American) baseball game going. Yes, the children won but personally I think the game was fixed... Thanks to the other parents who brought some other outdoor activities to occupy the children. Between eating the goodies and playing baseball the time was pretty much occupied, with the occasional slip inside the house to the computer (couldn't leave our of TI out, could we?) to demo some items and running up the modem for a bit. It was a really good day and I'm glad so many could show up! It was a nice time to meet and chat without that of dreaded meeting stuff! Hopefully, we'll have another one soon!

(To ease TONY BOWDEN's mind...) See TONY? I didn't even put one line in here about you showing up a whole day early for the Bar-B-Que, deccerts and all!

SEE EVERYONE AT THE NEXT MEETING:

SATURDAY JUNE 25, 1988 2:00 PM 13 ELM WALK

Now. onto other things...

Turn the page for more...

% WHAT'S NEW? % % On the Exchange Front %

A nice program from LINCOLN 99 COMPUTER CLUB (V6 #4 Apr 88) called OVERLAY STRIP. Have you run out of blank Key Strips? Well, now you can make your own. You should find an example of this within the newsletter. I typed this in from Lincoln's newsletter (along with the associated 'syntax errors' from typo's) but have it up and running and available on Disk in the Library. Ask for SE0039 (and get some FunTag Labels along with it!). Hopefully, you'll see an example of the FunTag Labels in this issue also.

From MID-SOUTH 99/4A USERS GROUP (V6 #4 Apr 88) the following news: McConn Software has announced the Avanti-99 Forth Card will soon be released. The card fits into the PEB and contains an RISC Novix NC4016 microprocessor. Program the chip directly in Forth with an increase in execution speed. It has 128K RAM and 16K ROM. % Great Lakes Software has an update to Certificate 99 (see "What's New at Tenex" section). % Asgard Software (always in the news) has Beyond Video Cness (Save and Load a game from disk; use joystick; screen dump to printer; list moves to printer). % Last issue we advertised their release of Calendar Maker 99 and other new programs. % According to Jack Riley of Myarc, the Myarc Hard Disk Controller Card was to begin shipment last month. % McWare Products of P. O. Box 2784, Fairfax, Virginia, USA 22031, has available a workbook for instructional purposes called "How To Control and Print from any Computer".

GUILFORD 99'er USER GROUP was kind enough to notate our First Birthday for EAR 99'ers in their Newsletter (V5 #5 May 88) - THANKS MACK! Their newsletter this month covers: Atrax Tracks; Rambling Bytes; Machine Code Mastery; Forth Tutorials Session 3; and CopyCats, about a program called Floppy Copy (\$25.95 + \$2.00 S/H from Softspot (P. O. Box 8786, Silver Spring, MD USA). Copies quickly in 24 seconds. Copies all

" WHAT'S NEW AT TENEX "

Print Wizard available at \$19.95 + S/H. Creates greeting cards, signs, banners and letterheads. Requires XB, 32K, and it's compatible with ProWriter, Epson FX, Gemini 10X or other Epson printers.

Go from a Bare TI-99/4A Console System to a FULL System for only 419.00 (marked down \$80.00). For British Members divide by the exchange rate for pound cost (approx \$226.00 + 17% vat and duty). Inis system has Parallel Printer Port, R9232 Port, 32K Memory Expansion, DSDD Disk Controller with a Half-Height Disk Drive and drive/cable package (with external box power supply) and Filler Plate (for that 2nd drive you'll purchase later!).

purchase later:..

Legends Version 1.1 is available at \$24.95 if you didn't purchase
the original version (which can be sent in with \$3.00 for the updated

Personal Auditor (\$44.95) is a professionally presented package that helps you audit your personal finances (AAGH!). A Home Accounting System helps you get a grasp on your personal accounts. (gulp!)

The TI Count System (originally priced at \$349.95) is NOW AT ONLY \$89.95! This is a truly professional accounting system with General Ledger, Accounts Receivable, Accounts Payable, Inventory, Payroll, and Mail System, with manuals.

nail bystem, with manuals.
Artist Enlarger (\$9.95) allows you to enlarge fonts and instances (designed for use with TI-Artist or Font Writer II). Enlarge, Stretch, distort horizontally and/or vertically, or reduce.

 \digamma icasso's Publisher (version 2.0) is available at \$19.95 plus there's a utilities disk at \$14.95. This seems to be the newest version and is also from Asgard Software. (Now I know why the author never

and is also from Asgard Software. (Now I know by the author hevel answered our letter regarding version 1.0 way back when...)

Elanner/Spread Roll paper (with no perforations) for those special banners you're making are available in White, Blue or Yellow at \$12.95 for one box, containing one color, 45" X 8 1/2" paper.

Half-Height Disk Drives are on sale at \$99.95 but still not as

inexpensive as B6 Micro at only \$79.95.

Certificate 99 is at \$19.95 with the Companion Disk at \$14.95. Creates extra borders and graphics and gives you the ability to save certificates as well as recall what you saved.

Bowling League Secretary at \$49.95 for keeping track of team scores, averages, and player improvement. Sets up a database for players, team names, scores, and averages. Print out all your data. Note this is the American version of Bowling...

Books: EXTENDED BASIC MANUAL available at \$8.95; A HACKER'S DELIGHT (TI Intern) at \$9.95; HARDWARE MANUALS FOR TI-99/4A at \$19.95; EDITOR/ASSEMBLER MANUALS at \$14.95; THE HACKER'S HANDBOOK at \$12.95; etc.

See the Tenex Catalog at the next meeting if you haven't gotten your new one yet (Spring 1988)!

> QVERLAY STRIP (Cut and Tape together) Perhaps glue on cardboard)

SHUFFLE YOUR	KEY TO TIE	LETTER VALUES	LETTER Bistribution	PASS	CHAMBE A LETTER
PLACE HORIZONTAL		PLACE VERTICAL	USE Dlank		SCRADDLE

PRINT-A-TAG / FUN-TAG LABELS (and variations of)

X X	Ť	,):	*	Ξſ	14	¥.	γ,	B 1	R1	THE	ΕÌ	/	* 5	*	*	* *
*			W	TH	1	•	L	M	₹ •	•	FF	106	1			*
*									EMI							*
*	*	*	*	*	1	1	*	*	*	1	*	*	*	*	*	X

SOCO HAPPY TAPPING! OCCE EAR 99'ers USER'S GROUP 13 ELM WALK LAKENHEATH, SUFFOLK ENGLAND IP27 90R G

GRAPHICS EDITOR (Examples of) (Over 100 graphics on disk)

GRAPHIC EDITOR



Version 2.0 ExBasic Autoload Freeware Documents



E/A OPT # 5 A-C

EAR 77'ERS UG



13 ELM WALK LAKENHEATH, SUFFOLK, ENGLAND TE27 908

SSGT D COPELAND MIC COST E

Page 5

East Anglia Region 99'ers User's Group

FOR SALE / WANTED

Use this space to advertise!

ITEM	# LEFT	COST (Each)
Full-Height Double-Sided Disk Drive	1	£ 50.00
Half-Height Single-Sided Disk Drive	Several	± 35.00
"Back-Issues"	_	± 1.00
Volume 1 Issue 1 May 1987	3	± 1.00 \$ 1.00
Volume i Issue 2 June 1987	3	# 1.00 \$ 1.00
Volume 1 Issue 3 July 1987	3	£ 1.00 \$ 1.00
Volume 1 Issue 4 Aug 1987	6	
Volume 1 Issue 5 Sept 1987	6	
Volume 1 Issue 6 Oct 1987	7	
Volume 1 Issue 7 Nov 1987	5	
Volume 1 Issue 8 Dec 1987	7	
Volume 1 Issue 9 Jan 1988	7	
Volume 1 Issue 10 Feb 1988	10	± 1.00 \$ 1.00
Volume 1 Issue 11 Mar 1988	11	± 1.00 \$ 1.00
Volume 1 Issue 12 Apr 1988	2	± 1.00
Volume 2 Issue 1 May 1988	18	± 1.00
Catalogs	26	± 1.00 • 1.00
Dual Cassette Leads	3	£ 2.00
TI Joysticks (Pair)	3	£ 5.00
Household Budget Management	1	± 2.00 \$ 2.00
Programmed Romox Cartridges	11	å 3.50
Cassette Tapes	1 of each	± 2.00

- # 16: Black Jack; Slot Machine; Tex-Thello; Battle Ship; Yahtzee; Vocabulary; Spelling; Space Patrol; Force 1; Spriter; Kingdom; Match-Up; Star Trek; Froggy; Build America; Gold Rush.
- Match-Up; Star Trek; Froggy; Build America; Bold Rush.
 (Requires Extended Basic Module)
 4: Subroutines and Demo's Typing; Moving Sprites; Graphics
 Definition; Title Screen; Black Jack; Phone Dialer; Small
 Letters; Computer Letters; ASCII Text; Display At and Accept
 At; Lower Letters; RND Sprite Dice; Computer Text Dice.
 (Requires Extended Basic Module) (Requires Extended Basic Module)
- 8: Quiz Master and Data Program Tape BB (Extended Basic Module)
 1: Preschool Block Letters; County Fair; Knight's Tour; Code Breaker; Christmas Music; Notes; Termite. (Basic and Extended Basic Mix)
- 2: Surcerer's Castle; Island Adventure; Four-In-A-Row; Haunted Sheakies. (Basic and Extended Basic Mix)
- 3: Programming Aids 1 (Extended Basic Module)
- 4: Artillery: Decipher: Puzzle 15; Flip Checkers (all fun!) (Extended Basic Module)
- 5: Basic Typist (Extended Basic Module)
- 6: Quiz Construction Set (Quizmake and Quiztake); Slither; Personal Loan Calculator; Bird Brain; Division Tutor; Jumping Ahead with Game Programming: LOGO Clones-(Basic, Extended Basic Module, Logo)
- # 7: Aztec Challenge (Extended Basic)

Teach Yourself Basic Chapters 1 - 5	2	*	1.00	1.00
Teach Yourself Basic Chapters 6 - 10	2	±	1.00	\$ 1.00
Teach Yourself Extended	Basic 1	*	1.00	\$ 1.00
TI Logo Sampler	1	*		\$ 1.00
Tunnels of Doom Pennies/Prizes; Quest	1			6 1.00
Scott Adams Adventures	1	t.	1.00	\$ 1.00
Voodoo Castle; Pirate	Adventure;	Adventure1 and;	Myster	y Fun

INDEX TO BACK ISSUES

Vi II May '87: Screen Dump Program; 99 Tips; Call Loads & Call Peeks Collated Listing; Word Processing CTRL U Commands; Buckaroo Banzai

Hints & Map.

Vi I2 June '87: Alpha-Blast; Strange Figures; Programming Tips;
SpaceStation Pheta Review; CTRL U Commands; Starcross Hints; Assembly Language Programming Tips.

VI 13 July '87: Crazy Climber; Contacts; Why T17; CTRL U Commands: Suspended Hints.

Vi I4 Aug '87: TI-99 Menu Loader; 32K to 16 Bit Bus; Alpha Lock Release; Ghost-Town Hints & Map; ZAP then ZERO!; Library Listing.

Vi IS Sept '87: Traffic Cop; Mini-Memory Tutorial; Library Listing; Burble & Minutes.

V1 I6 Oct '87: Burble & Minutes; Mini-Memory Tutorial; CodeBreaker; Q-Tips; Library Listing; Contacts; D.I.Y.; Zork II Hints & Map; Software Reviews.

V1 I6 Nov '87: Burble & Minutes; D.I.Y.; Witness Hints & Map: Spad XIII Review.

Vi IB Dec '87: Burble & Minutes; Pyramid of Doom & Map; Mini-Memory Tutorial; CTRL U Graphics; Reviews; Basic Program for the Holidays; Library Changes; Coupon Special.

V1 I9 Jan '88: TI-99/SG Upgrade; Mini-Memory Tutorial; New Releases & Reviews; Library Catalog.

VI IIO Feb '88: Minutes; Mini-Memory Tutorial; Terminal Emulator Program (Tickle Bone); Enchanter Hints & Map; Exchange Front News. VI III Mar '88: Minutes; In Remembrance; Mini-Memory Tutorial; Sorcerer Hints & Map; SuperSpace II Review; Terminal Emulator II Program

(Dunkman); Library Additions, VI 112 Apr '88: Minutes; Exchange Front News; Geneve Introduction; Mini-Memory Tutorial; Programming Graphics to Frinter; TI-Writer Tabs; For Sale.

V2 II May '88: Minutes; News; Disk Talk Tutorial; Zork III Hints & Map; Mini-Memory Tutorial; New Releases; Terminal Emulator Program (Rocky Robot in Learning to Count); Disk Sector Bit Map; FunnelWeb Flow Chart Enclosure.

> • FOR SALE / WANTED • + E.N.D. +

ADVENTUREMANIA ADAMITIS by: You-Know-Who Scott Adams Adventure Module > FOR ADULT'S ONLY < (Rated X)

Or depending on your views, for adult's who haven't grown up....

This Adult Adventure was written by Stephen Peacock of Jacksonville, Florida. and requires the use of the Scott Adams Adventure Module. Duplication and distribution encouraged, donation to the author. & Object of the adventure: You have been given a drug to make you forget who you are. You must find the account number of the safety deposit box which holds the antidote. There are nine (9) numbers to the account and you must figure out the order of the numbers. As you find them, remember them (although you won't find them in any particular order).

I'm only writing a hints column on this due to the number of requests for help (some all the way from the good ole' US of A). It was not my favorite and I must admit now that I had to get up from the computer and let hubby take over. You won't find the exact words in this hints column that'll get you your numbers for the entidote, but you'll get hints leading you in that direction. And, for this reason also, you will not find a Verb and Noun list (I won't reproduce them in the Newsletter). Mr. Peacock must have put alot of work into producing this APL Language adventure, but I figure his talents could have been put to better use... This is a personal opinion and Mr. Peacock need not take it personally, as I know alot of people are spending alot of time on this one. Anyway, on to the hints.....

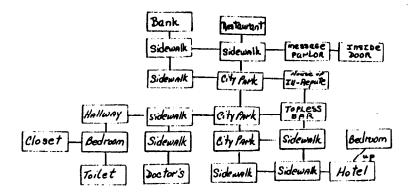
- Messages = 77; Objects = 49; Verbs = 44; Nouns = 74; Locations = 29; ou can carry 6 objects; Starting Location = 1; Length of Nouns/Verbs = 9
- " For those of you wondering what the "Word beginning with F" is, you've probably already figured out it's not what you think... There is, however, a vehicle that comes to your home when your home is on fire you'll hear the siren before it arrives...
- # Why not order something in the restaurant? The usual Chinese restaurant has Rice and Chicken. You never know, you might get a helpful Fortune Cookie. Try >SAY FOOD when the waitor asks what you want.
- σ That 'hippie' in the park with the Ghetto Jammer (Boom Box, whatever) is playing his stereo loud enough to hear. Why not try to <code>>Listen Radio?</code>
- # Most people go to the Doctor's Office for a physical examination. When the nurse asks what you want, why not ask for that annual checkup? Try > DAY FINGLEAL Remember (for the guys out there) this is not an ordinary office visit. While you're lying down, where's the only place to look?
- of Don't ask for help in this adventure... You won't like the
- a Don't leave the house without getting dressed; Don't leave the Doctor's Office with the examination gram; Don't leave the bank without telling the guard the number to the safety deposit box, etc. These are sure and easy ways to die (naturally causing a restart of the adventure).

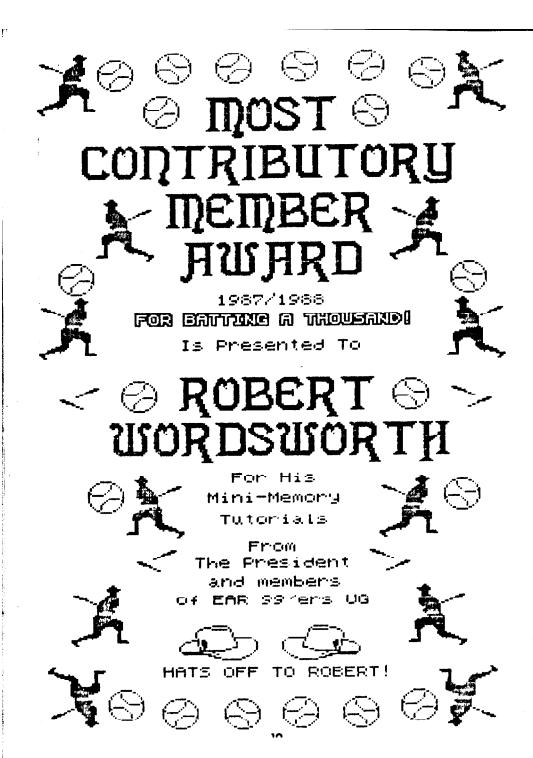
East Anglia Region 99'ers User's Group Page 8

- ** The newspaper gives you a hint on the safety deposit box numbers. Funny name for a group (Reverce) isn't it? (Although nowadays you never know...).
 - One way to see if the fish are ripe might be to >SMELL FISH.
 - → What size are your clothes?
- ਕੈ See the Map for further help that's all the hints you're getting! ਕੈ Next month ਕੈ The Garden Of Eden (Doors and Days) ਕੈ another Fairware Adventure using the Scott Adams Module... ਕੈ

→ ADVENTUREMANIA ADAMITIS →
→ E.N.D. →

for adult's "map" adventure





▶ COMPUTER KNOWLEDGE 4

- 45 THINGS YOU ALWAYS WANTED TO KNOW ABOUT 4
 COMPUTERS BUT WERE AFRAID TO ASK 4
- OR 4
 TITEMS ON COMPUTERS THAT YOU CAN FORGET ABOUT 4
 AS SOON AS YOU LEARN THE ANSWERS 4

Reference: The Hamlyn Book of Computer Questions & Answers by: Daniel Cohen

- ▶ 1 A computer that can handle a large volume of numbers.
- 2 A nineteenth-century English mathematician who designed an intricate system of gears and cogs that was called the Difference Engine (the first machine that could do more than simple calculations).
- $ilde{}$ 3 A form of mathematical logic developed by the nineteenth-century English mathematician, George Boole, where all problems can be solved by reducing them to a string of Yes or No, True or False choices.
- 4 Short for transfer resistance. A small 'sandwich' of material that can be used to control the flow of electricity.
- $\stackrel{ t }{ t}$ 5 A small board with the wiring for an electrical circuit 'printed' directly onto it.
- $^\flat$ $\,$ $\,$ $\,$ $\,$ $\,$ $\,$ $\,$ $\,$ Consists of transistors, other electronic components and the connections between them etched directly into a small piece of silicon.
- 7 The whole central processing unit of a computer etched into a single silicon chip.
- 8 A ruler with fixed and moving scales (a simple type of analog computer).
- 9 Comes from the Latin word meaning "two" and refers to two conditions
 the presence (1) or absence (0) of a pulse.
- ▶10 A contraction of the words Binary Digit.
- ${}^{\bullet}$ 11 A cluster of bits, usually eight. A grouping of eight bits needed to form a single letter or number.
- ▶12 16,384 bytes of memory.
- ▶13 An error or faulty instruction in a program or computer.
- \geq 14 The heart of any computer; the control center; the Gentral Processing Unit.
- P15 A storage unit, or memory unit, often used with small computers. Approximately the size of a 45 rpm record, made of flexible plastic, coated with a magnetic substance.
- ${}^{\flat}16\,$ A store of information often built right into the computer; can be read by the computer but it cannot be changed.
- ${\bf P}17$ The physical parts of the computer system, such as keyboard, display, printer, etc.

East Anglia Region 99'ers User's Group Page 11

- ▶18 A device that can create speech on instructions from a computer.
- Fig Essentially a computer's temporary memory; the computer stores programs and data fed into it from a tape or disk but when the computer is shut off all the memory is grased; when the computer is turned on again it's empty and ready to accept a new program and new data.
- >20 Programs used to control the computer.
- \triangleright 21 Any device connected to a computer that allows it to perform noncomputing tasks.
- P22 A printer having a row of needlelike pins that are individually fired against a ribbon as they move across the paper. Each pin produces a single dot, and patterns of the dots form individual letters.
- P23 Perhaps an output device such as a printer, but one adapted for the purpose of drawing diagrams, designs, pictures, patterns and symbols.
- $^{\rm P}$ 24 A printing element in many printers that produces fully formed solid characters or letters; the element roughly resembling a flower with a center and a host of 'petals' sticking out from the center.
- ▶25 A computerized machine. A machine directed by a computer.
- ▶26 A very small computer on wheels. Designed to find its way around a maze; it 'feels' its way around the walls with sensitive, infrared pulses and wires that stick out around it.
- F27 To represent the working of one system by another. In the world of digital computers physical systems are represented by mathematical models, and run through a computer to see how they will react.
- >28 A 'box' containing a ROM chip mounted on a circuit board. It has all the instructions for playing a game and the circuit board connects the ROM chip to the game console.
- ▶29 A type of game (usually) without pictures where the computer sets up a situation in words. The player responds by typing an instruction into the computer keyboard and the computer sets up another situation based on that response.
- ▶30 In 1972 a young computer technician built a game called Pong in his garage. (The name is half of Ping-Pong.) He went on to manufacture the game through his own company, called Atari.
- >31 The claim for being the first to create Pong (Ping-Pong) has been challenged by another computer engineer. He claims he had been working on this type of game since 1966 and in 1972 the game was brought out in a home video version by Magnavox.
 - F32 A small Japanese electronics company, Taito, Ltd., created this game that was so popular that the Bank of Japan had to triple its production of 100-yen coins, the denomination used to play the game.
 - >33 Originated in Japan, originally named Paku Paku, which roughly means "gobble gobble".
 - >34 Another Japanese game that became very popular. A literal translation of a Japanese phrase that means "stupid gorilla".

- *35 The name given to the greatest computer in the universe in the popular TV show and book International Designed by a race of hyperintelligent, pan-dimensional beings to answer the "ultimate question" about life, the universe, and everything.
- *36 The computer 'star' personality in the 1968 Stanley Kubrick film 2001: A Greec Odyssex-
- +37 The answer to Number 36 Decoded.
- $^{\rm h}38$. The $^{\rm h}$ of the computer that runs the StarShip Enterprise in Star Trek.
- ▶39 A combination of human being and machine.
- ▶40 Garbage In, Garbage Out.
- *41 The branch of computer science concerned with the possibility, the methods, and the implications of computers performing tasks in the same way the human brain does.
- P42 This pioneer artificial-intelligence researcher used the expression of calling the human brain a "meat computer".
- $^{\rm P}43$ A circle of enormous stones set up in England in prehistoric times. Some scientists believe that the people who built this used it as a kind of computer.
- $^{\circ}$ 44 The most successful ancient computing device which has been in use in China for over 2,500 years. Keeps track of calculations by moving beads up and down strings or bamboo rods.
- $^{\rm h}45\,^{\circ}$ A good candidate for the first "real" computer, assembled at Harvard University in 1944.

A.	Printed Circuit	В.	Microprocessor	c.	Speech Synthesizer
D.	Bit	E.	Bug	F.	Slide Rule
G.	Peripheral	н.	Floppy Disk	1.	Plotter
J.	Pac-Man	K.	Game Cartridge	L.	Robot
M.	Hardware	N.	Ralph Baer	0.	Byte
Р.	Binary	Q.	Deepthought	R.	IBM
s.	Dr. Marvin Mindky	Τ.	Cyborg	U.	Space Invaders
٧.	Micronouse	W.	Number Cruncher	X.	Stonehenge
٧.	HAL	Z.	CPU	1.	Boolean Logic
2.	Adventure Game	3.	ROM	4.	Abacus
5.	Charles Babbage	4.	Dot Matrix	7.	Daisy Wheel
8.	Software	9.	Computer	10.	RAM
11.	Nolan Bushnell	12.	Simulation	13.	G160
14.	16K	15.	Integrated Circuit	16.	Transistor
17-	Dankey Kong	10.	Mark I	19.	AI - Artificial Intelligence

Complete this, send in your answers, and (if correct), you get your choice of one (i) item from the Library FREE (whether it be on disk or cassette). (Members Only, although everyone is welcome to test their computer knowledge on this quiz!). Verify your answers next month!

► COMPUTER KNOWLEDGE 4 ► E.N.D. 4

MINI-MEMORY Part IX by: ROBERT WORDSWORTH

Last month we wrote a routine which was able to scroll the screen leftwards one column at a time. With some very small modifications, we can scroll two columns at a time.

The instruction whose address is >7E22;

MOVB #1+.@BU+32

moves the left-most byte of the buffer containing the line read from the screen to the end of the buffer, following the screen line. It also increments register 1 by one so that the buffer which the register is addressing is effectively "shifted" one byte to the right before we write the buffer contents back to the screen. As the line itself remains stationary in storage, relative to the buffer it moves leftwards one byte.

If we are going to scroll leftwards by two columns instead of one, the instruction above needs to move the two left-most bytes to the end of the buffer rather than just one, and register 1 needs to be incremented by two. This is easily accomplished simply by changing the MDVB (Move Byte) instruction to a MOV (Move Word):

MOV \$1+,@BU+32

Not only are two bytes moved rather than one, but the auto-incrementing, is by two, since MOV is a word instruction. There is room to hold two bytes at the end of the buffer because we reserved space for the buffer with a BSS 34 directive. The original reason for doing this was because the Line-by-Line Assembler keeps everything on word boundaries. If we had tried to reserve thirty-three bytes, we would in fact only have got thirty-two.

The only other modification we need to make is to the instruction at >7E2E: DEC 1 which, by decrementing register 1 by one, caused it to point back to the beginning of the buffer area ready to receive the next line from the screen. As we have "auto-incremented" register 1 by two, we now need to decrement it by two, and you'll be glad to know that we don't need to add an extra DEC instruction to do this since there is an instruction just for this purpose, Decrement by Two, coded in this case DECT 1. DECT is useful in "word" situations in the same way in which DEC is useful in "byte" situations. Since there is an INC instruction corresponding to DEC, you won't be surprised to know that there is also an Increment by Two instruction, INCT.

TNC, INCT, DEC, DECT are all examples of "single address" instructions, or Format VI instructions. You'll remember that these have just one operand, which can take any of the five forms of a "general address":

Page 14

Workspace Register
Workspace Register Indirect
Workspace Register Indirect Auto-increment
Symbolic Memory
Indexed Hemory

(as in the above example)
(for example #1)
(for example #1+)
(for example @BU or @>6030)
(which we haven't yet used)

Key these two changes in using "OLD":

AORG >7E22 MOV #1+,@BU+32 AORG >7E2E DECT 1

Note how the generated machine code at the left—hand side of the screen changes. When the MOVB is replaced by MOV, the generated D831 changes to C831. When the DEC is replaced by DECT, the generated 0601 is replaced by 0641.

Now run the routine as before, by calling from the BASIC program given in last month's issue. As you can see, the scrolling fairly zips along.

So far in this series of articles, we have covered some of the basics of Assembly Language programming. We've written some "pure" Assembly Language routines and also introduced the topic of calling machine code from BASIC. To some extent, there are two divergent paths from this point. One is to concentrate on calling machine code from BASIC. We have already learnt enough machine code instructions to be able to carry out some useful tasks, but still have to cover parameter passing. The other path is to concentrate on finding Assembly Language replacements for functions which would normally be carried out in BASIC, so that we can eventually write 100% machine code programs.

We'll be covering both options as we go along. One topic which is relevant to both approaches, and which we haven't yet looked at, is that of reading the keyboard. Of course TI BASIC has the INPUT statement, but, with it's scrolling and compulsory "beep", it hasn't the versatility of Extended BASIC'S ACCEPI AI. The rest of this article will concentrate on how to read the keyboard in Assembly Language, and develop an elementary INPUT routine.

Reading the keyboard in Assembly Language is very similar to using TI BASIC's CALL KEY. In fact the process is reminiscent of the "ACCEPT AT" subroutine supplied in "Frogramming Aids I", versions of which appear in many books on TI BASIC. The syntax of CALL KEY is:

CALL KEY(key-unit, return-variable, status-variable)

The key-unit specifies what we are going to read, the return-variable holds the ASCII code of the key pressed and the status-variable indicates whether or not a key has been pressed.

In Assembly Language the functions of these three parameters are fulfilled by three bytes in the CPU RAM PAD, the 236 bytes of high-speed Ram held inside the console.

The byte at >8374 is used to specify what we are going to read. To read the whole keyboard it is set to zero.

The byte at >8375 holds the ASCII code of the key pressed.

The byte at >837C, referred to as the GPL status byte, indicates whether or not a key has been pressed.

First we must set the byte at >8374 to zero to specify a scan of the whole keyboard. The easiest way to do this is to use the CLR (CLeaR) instruction. CLR is another "single address" instruction. The operand can have any format of general address. In this case we want to clear a byte in CPU RAM at a specific address so we use a symbolic address. In fact CLR acts on a complete word, but it doesn't matter in this case since initially the value in the byte at >8375 is immaterial. So we zero is the byte at >8374.

We must now call the Keyboard Scan routine (KSCAN) line. To do this we use a: BLWP 0>6020.

We will loop round obeying this instruction over and over again until we detect a key press. The ASCII value of the key pressed will then be in the byte at >8375.

To test whether a key has been pressed we need to check bit 2 of the Status Byte at >837C. The bits are numbered from the left, starting at 0. To test an individual bit we can use the Compare Ones Corresponding (COC) instruction. COC belongs to the small group of "Logical Instructions" (Format III instructions.)

Format III instructions have two operands. The first (source) operand is a general address. The second (destination) operand is a workspace register. To test whether bit 2 of the byte at >8375 has been pressed, we must first move it to a workspace register. We must then set up, as the source operand, a word containing the bit pattern 0010 0000 0000 0000. One of this can be done with the directive: BT DATA >2000.

The instructions would then be:

MOVB @>837C,1 COC @BT,1

The MOUB instruction moves the byte to the left—hand byte of register $\ 1.$ The contents of the right-hand byte doesn't matter because only bit 2 of the register is being checked.

The COC instruction sets the "equal bit" in the on-chip Status Register if all the bits in the destination operand which correspond to set bits of the source operand are themselves set to one. In this case only one bit, bit two, of the source operand &BT is set to one, so only bit two of the destination operand, register 1, has to be set for the equal bit in the Status Word to be set. To act on the setting of the Status Word's equal bit we use the conditional jump instructions JEQ (jump if equal bit set) and JNE (jump if equal bit not set) instructions. In this case we want to loop back if the equal bit isn't set so we would code:

JNE (address of BLWP @>6020)

As soon as a key has been pressed we will not jump back, and we will want to look at the ASCII value of the key. This is stored in the byte at >8375. As the byte at >8374 has already been cleared to signify a whole keyboard scan, we can regard the whole word at >8374 as containing the ASCII value of the key pressed. To compare the value with 13, the ASCII value for the enter key, we must move this word to a workspace register, since the Compare Immediate instruction can only be used on registers. If the enter key has indeed been pressed, we want to continue with the rest of the program, which in this case simply prints the whole input message. If the enter key hasn't been pressed we want to print the

East Anglia Region 99'ers User's Group Page 16

character corresponding to whatever key was in fart pressed and store it in the buffer. In fact our program is roughly equivalent to the BASIC program:

100 INPUT AS

To print the key pressed, we will use the VSBW routine. This requires the ASCII value of the character to be printed to be in the left-hand byte of register 1. At the moment however, it is in the right-hand byte. We need to swap the bytes in register 1. Fortunately there is an instruction to do just this: Swap Bytes, SWPB. This is another single address/Format VI instruction. We would code:

SMFB 1 BLWP @>6024

We are going to keep a count of the characters printed, so that when we come finally to print the whole input message using the VMBWrite routine, we have a length to pass to VMBW in register 2. Also we will want to make sure that this length is non-zero. We are going to keep this count in register 3, so we will want to initialize it to zero with a CLR 3 and instruction and increment it with INC 3.

Having printed an individual character as above, we shall move it to a buffer in CPU RAM, in preparation for the final printing of the complete message. For this purpose, we'll use the area labelled "BU" which we set up in last month's program. If we're using "OLD" and the Symbol Table is still intact, we can use this label without any problems, but if for any reason it's gone, we can set it up again with:

BU EQU >7DER

To move each character to the buffer, we can use the instruction:

MOVB 1, @BU(3)

This is an example of the fifth and last type of general address: Indexed Memory addressing. This is very similar to Symbolic Memory addressing with the additional feature that the contents of the Index Register, in this case Register 3, is added to the address value corresponding to the symbol. Initially, register 3 will contain zero, so the left-hand byte of register 1 will be moved to the first byte of the buffer. The next time, register 3 will contain one, so that the byte just printed will be moved to the second byte of the buffer, and so on.

On the next page you'll see the complete program...

```
XXXX XXXX
                ADRG >7E36
  7E36 02E0
               LWP1 >7088
                               Load Workspace pointer:
  7E38 7088
                               program entry point
  7E3A 04E0
               CLR @>8374
                               Prepare to scan whole keyboard.
  7E3C 8374
 7E3E 0200
                     0,>100
                               Position prompt at Row 9, Column 1.
 7E40 0100
 7E42 0201
                     1,>3F00
                              Prompt with question mark.
 7E44 3F00
 7E46 0420
               BLWP @>6024
                              Print prompt.
 7E48 6024
 7E4A 05C0
               INCT O
                              Set screen position to start
                              printing input.
 7E4C 04C3
               CLR 3
                               Initialize register 3 to hold
                               count of characters input.
                              This register is also used
                               as an index register.
 7E4E 0420 KS BLWP @>6020
                              Start reading keyboard.
 7E50 6020
 7E52 D060
               MOVB @>837C,1 Move status byte to register 1.
 7654 037C
 7E56 2060
               COC @BT, 1
                              Test bit 2 of status byte.
 7E58 7F88
 7E5A 16F9
                    KS Try again if not set. e>8374,1 Move word at >8374 to register 1.
               JNE
 7E5C C060
               MOV
 7E5E 8374
                              Left-hand byte is zero because
                              byte at >8374 has been cleared.
Byte at >8375 contains ASCII code
                              of key pressed.
7E60 0281
              CI
                    1,13
                              Enter key pressed?
 7E62 000D
 7E44 1308
               JEG
                              If yes, continue.
7E66 06C1
              SWPB 1
                              Swap bytes of register 1, placing
                              key code in left-hand byte.
7E6B 0420
              BLWP @>6024
                              Print character input on screen.
7E6A 6024
7E6C D8C1
              MOVB 1,880(3) Store character input in buffer,
7E6E 7DE8
                             indexed by character count.
Add 1 to character count.
7E70 0583
              INC
7E72 0580
              INC
                   O
                              Move screen pointer one byte
                              to the right.
7E74 10EC
                   KS
                             Loop back to read the next key.
                             Enter pressed
7E76 C083 CT MOV
                              Move length of text to be printed
                  3,2
                              to register 2.
7E7R 13EA
              JEO
                   KS
                              If no characters to be printed,
                             return to read keyboard again.
7E7A 0200
                   0.>200
                             Position to print input
7E7C 0200
7E7E 0201
7E80 7DE8
              LI
                   1, BU
                             Address of CPU bufter in reg 1
7EB2 0420
              BLWP @>6028
                             Print input on screen (VMBWrite).
7E84 6028
7EB6 10FF
              JMP
                             Endless loop
7EBB 2000 BT DATA >2000
                             Word with bit 2 set,
7EBA XXXX
              SYM
RESOLVED REFERENCES
BU-7DER KS-7E4E PT-7E88 CT 7E76
```

The program entry point is at >7E36. Let's choose a name for our program and add the name and the program entry point address to the REF/DEF table as described in last month's article.

Set the FFAM by entering:

7EBA XXXX AORG >701C 701C 7EBA DATA >7EBA

The location counter should now be >701E, the address of the LFAM field.

701E 7FDO DATA >7FDO

Now we must change the location counter to the new LFAM value and enter a new REF/DEF table entry there. So enter:

7FD6 7E36 DATA >7E36 7FD8 XXXX END

Save the whole of MiniMemory RAM to cassette with the Easybug S(ave) command, as usual, saving from 7000 to 7FFF.

We are now ready to run the program. This time we'll use the $\mbox{MiniMemory RUN option.}$

From the MiniMemory menu select option 2, "RUN". When asked for the program name, enter "INPUT". The screen should turn green and go blank, except for our prompt, a question mark. When we press a key, the character corresponding to that key should appear beside the prompt. When we press "enter" the complete message should appear on the screen. As the program is in an endless loop, the only way to stop it running is by switching the computer on and off.

Hope you're successful!

> MINI-MEMORY PART IX < > E.N.D. <

STOCK MARKET

You're gonna' like this one! Here's your chance to prove yourself in the Stock Market dame. Buy and sell your shares of stocks to make the most money. Go to Line 1490 and change the names of the companies for your own particular game. Buy Shares (outright or from another player), Sell Shares (outright, to the bank, to another player, etc.), Trade with other players, and make millions! This runs in Extended Basic (depress alpha-lock) and affords you hours of fun! Choose the number of weeks to play for a short or long game. Have fun (and remember me when you're rich!!).

```
100 REM *******
 110 REM #
 120 REM # STOCK
 130 REM #
                   MARKET
 140 REM 1
 150 REM ############
 160 REM
 170 REM AN EXTENDED BASIC PROGRAM
 180 REM
 190 CALL CLEAR :: DISPLAY AT(12,8): "STOCK MARKET" :: DISPLAY AT(24,2): "PRE
SS ANY KEY TO CONTINUE" :: GOSUB 1460 :: CALL CLEAR

200 DPTION BASE 1 :: DIM S(6),5*(6),F(10,6),FC(10,2),CH(6,2):: RANDOM1ZE

210 CALL CHAR(33,"000000FFFF",95,"181818181818181818"):: CALL SCREEN(2):: FO

R Z=1 TO B :: CALL COLOR(Z,16,5):: NEXT Z
220 RESTORE 1490 :: FOR Z=1 TO 6 :: READ S$(Z):: CH(Z,1)=INT(RND$4+1):: CH
(Z, 2)=INT(RND#7)-3 :: NEXT Z
230 CALL CLEAR :: DISPLAY AT(1,1): "HOW MANY PLAYERS (1-10)?2" :: ACCEPT AT
230 CHLL CLERK II DISPLAY HI(I).I):"HUM THAY PLAYERS (1-10)/2" :: HULEP) HI
(1,25)SIZE(-2)VALIDATE(DIGIT):NP :: IF NP<1 OR NP>10 THEN 230
240 FOR Z=1 TO NP :: DISPLAY AT(Z+3,1):"PLAYER #";STR$(Z);"'S NAME:" :: AC
CEPT AT(Z+3,19)SIZE(8):NAM$(Z):: NEXT Z
250 FOR Z=1 IO NP :: IF NAM$(Z):"" THEN NAM$(Z):"PLAYER*&STR$(Z)
260 NEXT Z
270 DISPLAY AT(16,1):"HOW MANY MEEKS(2 OR MORE)?":"25" :: ACCEPT AT(17,1)V ALIDATE(DIGIT)SIZE(-3):A$ :: IF A$<"2" THEN 270 ELSE NW=VAL(A$) 280 CALL CLEAR :: FOR Z=1 TO 6 :: S(Z)=INT(RND#40+10):: FOR Z1=1 TO NP ::
P(Z1,Z)=0 :: NEXT Z1 :: NEXT Z
290 FOR Z=1 TO NP :: PC(Z,1)=5000 :: PC(Z,2)=0 :: NEXT Z :: FN=1 :: W=1 ::
 GOSUB 1400
300 GDSUB 1460 :: IF K=66 DR K=83 THEN 320 ELSE IF K=84 THEN 420 ELSE IF K
=76 THEN 970 ELSE IF K=78 THEN 1150
310 IF K#80 THEN 1290 ELSE CALL SOUND(1,110,0):: GOTO 300
320 M=K :: GDSUB 1470 :: DISPLAY AT(22,1):"WHICH STOCK (1-6)?" :: GOSUB 14 60 :: IF K<48 OR K>54 THEN 320 ELSE SN=K-48
330 IF SN=0 THEN GOSUB 1430 :: GOTO 300
340 IF S(SN)=0 THEN DISPLAY AT(23,1):"THAT STOCK IS BANKRUPT" :: CALL D(50
0):: 60TO 320
350 DISPLAY AT(22, 20):58(SN): "HOW MANY SHARES?" :: ACCEPT AT(23, 17) VALIDAT
E(DIGIT): 75 :: IF 75="" THEN 350 ELSE NOS=VAL (26)
360 IF NOS=0 THEN GOSUB 1430 :: GOTO 300
370 IF M=83 THEN 400
380 COST-S(SN) #NOS :: CHRG=INT(COST#.006):: IF COST+CHRG>PC(PN,1)THEN DISP
LAY AT (24,1): "NOT ENOUGH MONEY" :: CALL D(500):: GOTO 320
390 DISPLAY AT(24,1): "PURCHASE COMPLETE: "; COST+CHRG :: PC(PN,1)=PC(PN,1)-C
OST-CHRG :: P(PN,SN)=P(PN,SN)+NDS :: CALL D(500):: GDTD 1290
400 IF NDS>P(FN,SN)THEN DISPLAY AT(24,1): "NDT ENDUGH SHARES" :: CALL D(500
):: K≖M :: 60TO 320
```

```
410 DISPLAY AT (24,1): "STOCK IS SCLD: "; NOS#S(SN):: PC(PN,1) =PC(PN,1) +NOB#8(SN):: P(PN,SN) =P(PN,SN) -NOS :: CALL D(750):: GOTO 1290
420 CALL CLEAR :: DISPLAY AT (1,1): "1) TRADE WITH OTHER PLAYERS ": 1"2) TRADE FOR ANOTHER STOCK ": : "3) SELL TO THE BANK"
 430 DISPLAY AT(7,1): "ENTER YOUR CHOICE": "OR PRESS O TO RETURN"
450 IF K<48 OR K>51 THEN 440 ELSE IF K=48 THEN GOSUB 1400 :: GOTO 300 ELSE
  ON K-48 5010 460,790,880
460 IF NP±1 THEN DISPLAY AT(10,1):"YOU'RE THE ONLY PLAYER" :: CALL D(750): 60T0 420
470 CALL CLEAR :: GOSUB 960 :: DISPLAY AT(10,1): "TRADE WITH THESE PLAYERS:
480 X=11 :: Y=1 :: FOR Z=1 TO NP :: IF Z<>PN THEN DISPLAY AT(X,Y):Z;NAM&(Z):: X=X+1_:: IF X>15 THEN X=11 :: Y=15
490 NEXT Z
500 DISPLAY AT(17,1):"WHICH PLAYER?" :: ACCEPT AT(17,15)VALIDATE(DIGIT)SIZ
E(2):A$ :: IF A$="" THEN 500 ELSE TN=VAL(A$):: IF TN=0 THEN 420 510 IF TN>NP OR TN=PN THEN 500
520 FDR Z=1 TD 6 :: DISPLAY AT(Z+2,12):USING "### ###### ######":S(Z),P(PN
 . Z) . P(TN. Z) :: NEXT Z
530 DISPLAY AT (2.1): "STOCK
                                                                         VAL PL#":PN:TAB(23):"PL#":TN
540 DISPLAY AT(19,1):"1) SELL TO "; NAM$(TN):"2) BUY FROM "; NAM$(TN):"YOUR
CHOICE?" :: GOSUB 1460 :: IF K<49 OR K>50 THEN 540
550 IF K=50 THEN 670
560 CALL HCHAR(19.1.32.192):: DISPLAY AT(19.1): "SELL WHICH STOCK?"
570 GOSUB 1460 :: IF K<48 OR K>54 THEN 570 ELSE SN=K-48 :: IF SN=0 THEN 42
0 ELSE IF P(PN,SN)=0 OR S(SN)=0 THEN 570
580 DISPLAY AT(19,19):5$(SN):"HOW MANY SHARES?"
590 ACCEPT AT(20,19)VALIDATE(DIGIT):2$ :: IF Z$="" THEN 590 ELSE NOS=VAL(Z
$):: IF NOS>P(PN, SN) THEN CALL SOUND(1, 110, 0):: GOTO 590
600 IF NOS=0 THEN 420
610 DISPLAY AT(21,1): "SELLING PRICE?" :: ACCEPT AT(21,16) VALIDATE(DIGIT):A

* :: IF A*="" THEN 610 ELSE SP=VAL(A*)

620 IF SP<S(SN)/2 OR SP>S(SN)*2 THEN 610
630 IF SP#NOS>PC(TN,1)THEN DISPLAY AT(22,1):NAM*(TN); " CAN'T AFFORD IT" ::
CALL D(500):: GOTO 560
640 DISPLAY AT (22,1):NAM*(TN):" DD YOU AGREE":"(Y/N)?" :: ACCEPT AT (23,7) V ALIDATE ("YN") SIZE (1):AG* :: IF AG*="N" THEN 560
650 S(SN)=S(SN)+INT((SP-S(SN))/10)
660 \text{ PC(PN,1)=PC(PN,1)+NOS*SP } :: PC(TN,1)=PC(TN,1)-NOS*SP } :: P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,SN)=P(PN,S
SN)-NOS :: P(TN,SN)=P(TN,SN)+NOS :: GÓTO 420
670 CALL HCHAR(19,1,32,192):: DISPLAY AT(19,1):"BUY WHICH STOCK?"
680 GOSUB 1460 :: IF K<48 OR K>54 THEN 680 ELSE SN=K-48 :: IF SN=O THEN 42
0 ELSE IF P(TN, SN) =0 OR S(SN) =0 THEN 680
690 DISPLAY AT(19, 18):54(SN):"HOW MANY SHARES?"
700 ACCEPT AT(20, 19) VALIDATE(DIGIT):A8 :: IF A8="" THEN 700 ELSE NOS=VAL(4
$):: IF NOS>P(TN,SN)THEN CALL SOUND(1,110,0):: GDTO 700
710 IF NOS=0 THEN 420
720 DISPLAY AT (21,1): "SELLING PRICE?" :: ACCEPT AT (21,16) VALIDATE (DIGIT):A

$ :: IF A$="" THEN 720 ELSE SP=VAL(A$)
730 IF SP=0 THEN 420
740 IF SP<S(SN)/2 OR SP>S(SN)#2 THEN 720
750 IF SP$NOS>PC(PN,1)THEN DISPLAY AT(22,1):NAM*(PN);" CAN'T AFFORD IT" :: CALL D(500):: 60T0 670
760 DISPLAY AT(22,1):NAM$(PN);" DO YOU AGREE":"(Y/N)?" :: ACCEPT AT(23,7)V
ALIDATE("YN")SIZE(1):AG$ :: IF AG$="N" THEN 670
770 S(SN)=S(SN)+INT((SP-S(SN))/10)
780 PC(TN,1)=PC(TN,1)+NOS$SP :: PC(PN,1)=PC(PN,1)-NOS$SP :: P(TN,SN)=P(TN,
SN)-NOS :: P(PN,SN)=P(PN,SN)+NOS :: 60T0 420
790 CALL CLEAR :: GOSUB 960
BOO DISPLAY AT(15,1):"TRADE WHICH STOCK?" :: GOSUB 1460 :: IF K<48 OR K>54 THEN 790 ELSE IF K=48 THEN 420 ELSE SN=K-48
810 IF P(PN, SN) =0 THEN 800
```

```
820 DISPLAY AT(15,20):S$(SN):"FOR WHICH STOCKO" :: GOSUB 1460 :: IF K<48 O
 R K>54 OR K-48=SN THEN 820 ELSE IF N=48 THEN 420 ELSE TS=K-48
830 DISPLAY AT(16,18):5$(TS)::"HOW MANY SHAKES OF ";5$(SN):: ACCEPT AT(19,1)VALIDATE(DIGIT):A$ :: IF A$="" THEN 830 ELSE NOS=VAL(A$)
  840 IF NOS>P(PN,SN)THEN 830
 850 V=S(SN)*NOS :: NNS=INT(V/S(TS)):: TTB=INT((V/S(TS)-NNS)*S(TS))
860 DISPLAY AT(20,1):"IRADE";NOS;" SHARES OF ":$$(SN);" FOR";NNS;" SHARES
OF":$$(TS):"TRANSFER *";TTB;" TO YOUR ACCOUNT"
 870 P(PK,SN)=P(PN,SN)-NOS :: P(PN,TS)=P(PN,TS)+NNS :: PC(PN,1)*PC(PN,1)+TT
 B :: CALL D(750):: GOTO 420
 880 CALL CLEAR 11 GUSUB 960
 890 DISPLAY AT(11,1): "CASH IN WHICH STOCK?" :: GOSUB 1460 :: IF K<48 OR K>
 54 THEN 880 ELSE IF K=48 THEN 420 ELSE SN=K-48
 900 IF P(PN, SN)=0 THEN 890
910 DISPLAY AT(11, 22):50 (SN): :"HOW MANY SHARES?" :: ACCEPT AT(13, 18) VALID
 ATE (DIGIT): AS :: IF AS="" THEN 910 ELSE NOS=VAL (AS)
 920 IF NOS>P(PN, SN) THEN 910
 930 OFR=INT(RND$10)-5 :: DISPLAY AT(15,22):"THE BANK OFFERS YOU $";S(SN)+0
 940 DISPLAY AT(17,1):"DO YOU AGREE (Y/N)?" :: GOSUB 1460 :: IF K=78 THEN 4
 20 ELSE IF K(>89 THEN 940
 950 P(PN,SN)=P(PN,SN)-NOS :: PC(PN,1)=PC(PN,1)+NOS*(S(SN)+DFR):: GOTO 420
940 FOR Z=1 TO 6 :: DISPLAY AT(Z+2,1):USING 1480:Z,5*(Z),S(Z),P(PN,Z):: NE
 XT Z :: RETURN
 970 CALL CLEAR :: DISPLAY AT(1,1):"1) TAKE OUT A LOAN": :"2) PAY BACK A LO
 AN":
          "3) COMPOUND INTEREST ON A":"
                                                       FUTURE LOAN"
980 DISPLAY AT (8,1): "ENTER YOUR CHOICE": "UN PRESS O TO RETURN TO MENU"
990 DISPLAY AT (8,1): "ENTER YOUR CHOICE": "UN PRESS O TO RETURN TO MENU"
990 GOSUB 1460 :: IF K<48 OR K>51 THEN 990 ELSE IF K=48 THEN GOSUB 1400 ::
GOSUB 1430 :: GOTO 300 ELSE ON K-48 GOTO 1000,1040,1110
1000 V=0 :: FOR Z=1 TO 6 :: V=V+P(PN,Z) &S(Z):: NEXT Z :: LIM=MAX(5000-PC(P
 N, 2), V+PC(PN, 1)-PC(PN, 2))
1010 DISPLAY AT(11,1): "HOW MUCH MONEY DO YOU WANT TO BORROW?" :: ACCEPT A T(12,12) VALIDATE(DIGIT): A$ :: IF A$="" THEN 1010 ELSE BOR=VAL(A$)
1020 IF BOR+PC(PN,2)>LIM THEN DISPLAY AT(14,1):"YOU'RE OVER YOUR LIMIT":"M AXIMUM LOAN AMOUNT 10";LIM :: CALL D(000):: GUIU Y/O
1030 DISPLAY AT(14,1):"LOAN IS OK" :: PC(PN,1)=PC(PN,1)+BOR :: PC(PN,2)=PC
(PN,2)+BOR :: CALL D(250):: GOSUB 1400 :: GOTO 300
1040 IF FC(PN,1)=0 THEN DISPLAY AT(11,1): "YOU DON'T HAVE ANY MONEY" :: CAL
L D(750) .. GOTO 970
1050 IF PC(PN,2)=0 THEN DISPLAY AT(11,1):"YOU DON'T HAVE ANY DEBTS" :: CAL
L D(750):: 6010 970
1060 DISPLAY AT(11,1): "AMOUNT OF PAYMENT?" :: ACCEPT AT(11,20) VALIDATE(DIG
 IT):A$ :: IF A$=""
                           THEN 1060 ELSE LP=VAL(A$):: IF LP(0 THEN 1060
1070 IF LP=0 THEN 970
1080 IF LP>PC(FN, 1) THEN DISPLAY AT(13, 1): "YOU DON'T HAVE THAT MUCH" :: CAL
L D(500):: GOTO 970
1090 DISPLAY AT(13,1): "PAYMENT RECEIVED": "THANK YOU" :: PC(FN,1)=PC(PN,1)-
MIN(LP, PC(PN, 2)):: PC(PN, 2) =PC(PN, 2) -MIN(LP, PC(PN, 2))
1100 CALL D(500):: GOSUB 1400 :: GOTO 300
1110 DISPLAY AT(11,1): "AMOUNT FOR LOAN CALCULATION?" :: ACCEPT AT(12,1)VAL TRATE (DIGIT): A4 .: IF A4-" THEN 1110 ELSE BOR=VAL(A5)
1120 TI=0 :: B=BOR :: FOR Z=W TO NW :: I=INT(Bt.005):: B=B+I :: TI=TI+I ::
1130 DISPLAY AT(14,1): "WEEKS REMAINING:
                                                            "; NW-W: "LOAN AMOUNT: "; TAB (20); BO
R:"INTEREST:";TAB(20);TI;:"WEEKLY CHARGE;
1140 DISPLAY AT(18,1):"LOAN BALANCE:
                                                                   "; INT (BURE, 005)
                                                            "; B: : "PRESS ANY KEY TO CONTINUE
":: 60SUB 1460 :: G0SUB 1400 :: G0TO 300
1150 60SUB 1470 :: DISPLAY AT(22,1):"WANT TO END YOUR TURN(Y/N)?"
1160 60SUB 1460 :: IF K=78 THEN GOSUB 1430 :: GOTO 300 ELSE IF K<>89 THEN
 1160
1170 PN=PN+1 :: IF PN:NP THEN PN=1 :: W=W+1 ELSE GOSUB 1410 :: 60T0 300
1180 IF WONW THEN 1330
1190 FOR 7=1 TO NP ++ PC(Z,1)=FC(Z,1)=INT(PC(Z,2)#.005):: IF PC(Z,1)<0 THE
N PC(Z,2)=PC(Z,2)-PC(Z,1):: PC(Z,1)=0
```

```
1200 NEXT Z :: FOR Z=1 TO 6 :: C=INI(RND#20/CH(Z,1)+CH(Z,2))#5GN(RND#10)-3
  :: S(Z)=S(Z)+C
1210 JE 5(7) (#150 THEN 1230
1220 S(Z)=INT(S(Z)/2+1):: FOR ZZ=1 TO NP :: P(ZZ,Z)=P(ZZ,Z)*2 :: NEXT ZZ
1230 IF S(Z)>0 THEN 1250 ELSE S(Z)=0
 1240 FOR ZZ=1 TO NP :: P(ZZ,Z)=0 :: NEXT ZZ
1250 NEXT Z
 1260 IF INT(W/4)<>W/4 THEN 1280
1270 FOR Z=1 TO 6 :: CH(Z,1)=INT(RND#4+1):: CH(Z,2)=INT(RND#7)-3 :: NEXT Z 1280 GOSUB 1410 :: GOTO 300
1290 TOT=0 :: DISPLAY AT(4,1):"STOCK VALUE SHARES WORTH" :: FOR Z=1 TO
1300 DISPLAY AT(Z#2+4,11):USING "### ###### #######":S(Z),P(PN,Z),P(PN,Z) * S(Z):: TOT=TOT+P(PN,Z) *S(Z):: NEXT Z
1310 TOTAL=TOT+PC(FN,1)-PC(FN,2):: DISPLAY AT(22,1): "CASH: ":PC(PN,1):TAB(1
6); "INV: "; TOT: "LOAN: "; PC (PN, 2); TAB (16); "NET: "; TOTAL
1320 DISPLAY AT(24,1): "PRESS ANY KEY TO CONTINUE" :: GOSUB 1460 :: GOSUB 1
410 :: GBTB 300
1330 CALL CLEAR
1340 WIN=1 :: HV=0 :: FOR Z=1 TO NP :: TT=0 :: FOR ZZ=1 TO 6 :: TT=TT+(P(Z,Z)) +S(ZZ)):: NEXT ZZ :: TT=TT+PC(Z,1)-PC(Z,2)-5000
1350 IF IT>HV THEN HV=TT :: WIN=Z
1360 DISPLAY AT(Z+3,1):NAM$(Z);" GAINED $";TT :: NEXT Z
1370 DISPLAY AT(20,1):"THE WINNER IS: "; NAM* (WIN)
1380 DISPLAY AT(24,1):"PLAY AGAIN(Y/N)?";:: GOSUB 1460 :: IF K=78 THEN END
ELSE IF K<>89 THEN 1380
ELSE IF K 287 INEN 1380

1390 FOR Z=1 TO NP :: FOR ZZ=1 TO 6 :: P(Z,ZZ)=0 :: S(ZZ)=0 :: NEXT ZZ ::

PC(Z,1)=5000 :: PC(Z,2)=0 :: NEXT Z :: W=1 :: PN=1 :: GOTO 280

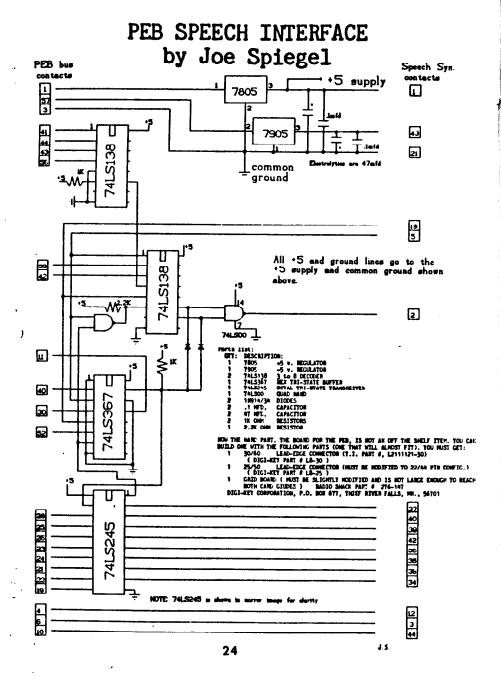
1400 CALL CLEAR :: FOR Z=3 TO 21 STEP 2 :: CALL HCHAR(Z,1,33,32):: NEXT Z

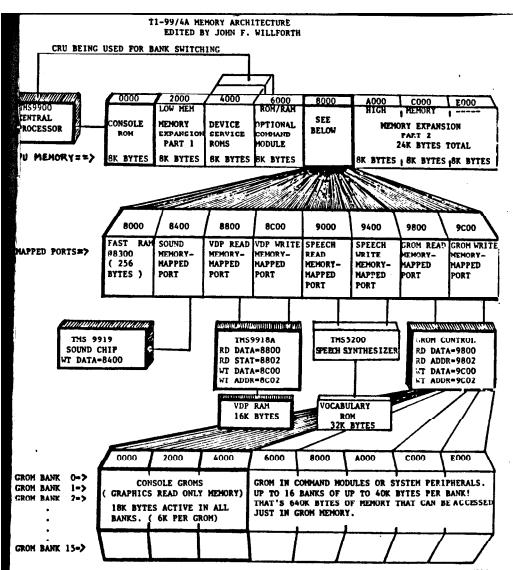
:: DISPLAY AI(1,8): "WALL STREET" :: CALL VCHAR(1,31,95,96)

1410 DISPLAY AT(4,1): "STOCK VALUE SHARES";:: FOR Z=1 TO 6 :: DIS
                                                  VALUE
                                                                SHARES"::: FOR Z=1 TO 6 :: DIS
PLAY AT (212+4, 1): USING 1480: Z, S$(Z), S(Z), P(PN, Z):: NEXT Z
1420 HI=0 :: L0=200 :: FOR Z=1 TO 6 :: HI=MAX(HI,S(Z)):: L0=MIN(L0,S(Z)):: NEXT Z :: DISPLAY A1(20,1):"HIGH:";HI;TAB(18);"LOW:";LU
1430 DISPLAY AT(22,1):"MENU: B)UY S)ELL":" T)RADE L)OAN":"
       N)EXT
                     P) ORTFOLIO*:
1440 DISPLAY AT (2,1): NAM$ (PN); TAB(19); "WEEK: "; W
1460 CALL KEY(0,K,ST):: IF ST<>1 THEN 1460 ELSE CALL SDUND(50,3000,5):: RE
TURN
1470 DISPLAY AT(22,1):RPT$(" ",84);:: RETURN
1480 IMAGE # ####### ###
1490 DATA US STEEL, PAN AM, FORD, SANYO, XEROX, AT&T
1500 SUB D(TDV)
1510 FOR TD=1 TO TDV :: NEXT TD :: SUBEND
```

> STOCK MARKET < > E.N.D. <

Page 23





NOTE: THE HEAVY LINES INDICATE FEATURES INCLUDED WITHIN THE CONSOLE. I HOPE THAT THIS MAP WILL BE OF SOME USE IN CLEARING UP THE MEMORY SCHEME THAT WAS DEVELOPED BY II TO MAKE MAXIMUM USE OF THE "CROM", AND THE ADDRESSING LIMITATIONS OF THE CPU (32K WORDS). FROM THIS DIAGRAM YOU CAN SEE, HOWEVER, THAT THE TI-99/4A HAS HAD THE POTENTIAL SINCE IT'S INCEPTION, TO BE A GRE. COMPUTING MACHINE.

ΩΩ CASSETTE CORNER ΩΩ

 Ω GONE FISHING! Ω Ω from: 919 Users Group Ω Ω February 1988 Ω Ω Extended Basic Module Ω

```
100 REM +++ GONE FISHING +++
100 REM +++ GUNE TAGENTS
110 REM +++ STIMSIM +++
120 REM SET PROBABILITIES AND DENSITY
130 DIM P(8,8),D(8,8)
140 RANDOMIZE
150 CALL CLEAR
160 CALL SCREEN(13)
170 FOR I=1 TO 8
180 FOR J=1 TO B
190 P(I,J)=.7#RND
200 D(1,J)-INT(RND#5+1)
210 NEXT J
220 NEXT I
230 P(1,1)=0
240 LB≠0
250 R=1
260 C=1
270 REM MAIN LOOP
280 FOR T=0 TO 6 STEP .1
290 PRINT
300 IF (RND>P(R,C))+(D(R,C)<1)THEN 310 ELSE 330
                  NO BITES!": : :
310 PRINT "
320 GOTO 390
330 N=INT(RND#D(R,C)+1)
340 W=INT(RND#R#C)+1
350 LB=LB+N#W
360 PRINT
370 PRINT " YOU CAUGHT": N: "FISH."
370 PRINT " YOU CHUSHI";N;"Fish;
380 PRINT " EACH WEIGHING";W;"LBS.;"
390 PRINT " AT LOCATION";C
400 PRINT : : "TOTAL LBS. THIS TRIP IS";LB;
410 PRINT : : :
420 REM UNEXPECTED EXPERIENCES
430 IF RNDCT/60 THEN 440 ELSE 460 440 PRINT "STORM--LOST 1/2 HOUR": : :
450 T=T+.5
460 J=INT(100#RND)+1
470 IF J>4 THEN 490
480 DN J GOSUB 820,910,990,1050
490 FRINT " YOU HAVE FISHED FOR";T; "HOURS.": : : : : 500 INPUT "MOVE(N,S,E,W,F,B)? ":M$
510 IF M$="E" THEN 520 ELSE 530
520 C=C+1
530 IF M$="N" THEN 540 ELSE 550
540 R=R-1
550 IF M$="W" THEN 560 ELSE 570
560 C=C-1
570 IF M$="S" THEN 580 ELSE 590
580 R-R+1
590 IF M$="B" THEN 600 ELSE 610
600 GOTO 120
610 IF (R<1)+(R>B)+(C/1)+(C>B) THEN 620 ELSE 640
620 PRINT "GROUNDED--SUNE!!": : : "ALL HAS BEEN LOST!!!!": : : :
```

```
630 GOTO 770
640 IF (R=1)*(C=1)THEN 700 ELSE 650
650 NEXT T
660 PRINT "TIME UP! THE SUN HAS SET.": : :
670 PRINT "1/2 YOUR CATCH HAS SPOILED.": : : :
680 LB=LB/2
690 REM SUMMARY OF TRIP
690 REM SUMMARY OF TRIP
700 IF T=0 THEN 710 ELSE 730
710 PRINT "STILL AT DOCK": : :
720 GOTO 150
730 PRINT "YOU ARE BACK AT THE DOCK"
740 PRINT "AFTER"; T; "HOURS OF FISHING."
750 PRINT "CLEAN": LB; "LBS. OF FISH."
760 PRINT "YOU RATE"; INT(LB/5); "AS A FISHERMAN."
770 PRINT : :
780 INPUT "ANOTHER TRIP? ":X$
790 IF X$="Y" THEN 140
BOO END
BIO REM SUBROUTINE
820 IF R+C<9 THEN 900
830 PRINT "FISH SCARED OFF BY SHARK."
840 PRINT "NOT BITING AS OFTEN."
850 FOR 1=1 TO B
860 FOR J=1 TO 8
870 P(I.J)=P(I.J)-.1
880 NEXT J
890 NEXT I
900 RETURN
910 PRINT "SEA GULLS ATE SOME OF YOUR BAIT."
920 PRINT "CATCH WILL BE SMALLER THIS TRIP."
930 FOR I=1 TO 8
940 FOR J=1 TO 8
950 D(I.J)=D(I.J)-1
960 NEXT J
 970 NEXT I
 980 RETURN
 990 PRINT "WATER SPOUT DISPLACES YOU."
1000 R=INT(8*RND+1)
 1010 C=INT(B#RND+1)
 1020 PRINT "YOU ARE NOW AT LOCATION"; R; C
1030 T=T+.2
 1040 RETURN
 1050 PRINT "YOU CAUGHT A 50-LB. SHARK."
 1060 LB=LB+50
1070 FRIN) "TOTAL LBS. THIS TRIP IS";LB;"."
 1080 RETURN
```

ΩΩ GONE FISHING! ΩΩ ΩΩ E.N.D. ΩΩ

INCOME Library Tapes and Disks Income \$ 58.50 Subscriptions Income \$ 9.00 Miscellaneous Income \$ 9.00 Modem Income This Month \$ 18.00 Sales Income This Month \$ 135.00

EXPENDITURES:

> \$2 END-OF-MONTH TOTAL 28 \$ 390.34

Petty Cash \$ 26.09 # 6.43

Checkbook Balances: American \$ 169.40

British & 52.35

Checks for Deposit: 1 @ # 34.00 1 @ \$ 23.20

Exchange flate at \$1.85/\$1.90=%1.00

Estimated Postage Expenses \$ 30.00
Photocopying Expenses = \$ 58.73

Vol 2 Issue 1 '88 ~ 14 originals @ 90 = 1260 @ .02599 = \$ 32.74

FunlWeb Insert ~ 2 originals @ 100 = 200 @ .02599 = \$ 5.20

Catalog Printing ~ 8 originals @ 100 = 200 @ .02599 = \$ 20.79

ESTIMATED INCOME:

Subscription Renewals (June 1988) \$ 50.00 Library Income (June 1988) \$ 35.00

>>>> NEXT MEETING: <<<<<

DAY: SATURDAY
DATE: JUNE 25, 1988
TIME: 2:00 FM
PLACE: 13 ELM WALK

SEE YOU THERE!

Buffer Full . . . E N D