

75C per issue

A Letter from the Editor

"The flowers, the gorgeous, mystic multi-colored flowers are not the flowers of life, but people, yes people are the true flowers of life: and it has been a most precious pleasure to have temporarily strolled in your garden."

Lord Buckley

No, this isn't a poetry journal or philosophical tract, but we do feel that Lord Buckley elucidates the concept behind the Processor Technology newsletter: its contents will be the kind of information that allows you to expand the creative applications of your personal computing system.

ACCESS is dedicated to the creation of a personal dialog between us at PTC and you, the people using our products, because we believe that it's important for you to understand your present system well enough to use it effectively, before you become trapped in the "bigger means better" syndrome.

Hence there *will* be no advertising of any kind in ACCESS, and no thinly disguised pitches aimed at getting you to part with your hard earned money for the unneccessary duplication of existing hardware.

Each issue will contain articles, engineering/application bulletins, and software (PROGRAM) listings which we think will help you get more fun out of owning your computer.

All we expect from you in return is what's known in technical and psychological circles as feedback. If you have devised a neat hardware modification or written a nifty program, send it to us and you'll get no financial remuneration. You'll just get the glory of seeing your name in print. Very spiritual, eh what?

Aram Attarian II

Subscription Information

ACCESS will be published every six weeks. This first issue is being sent free to all of our customers. If you like what you see, we hope you'll send us \$4.00 for a year's subscription so we can keep the info coming. Write to us at Processor Technology, 6200 Hollis Street, Emeryville, CA 94608.

One to One Communication

Open two-way communication is our goal, and it would be great to have unlimited time to rap with you over the phone. But we can't manage that logistically, so we are instituting procedures that will get your questions answered efficiently, without eliminating the personal touch.

You can get through to our engineers and technical staff any time between 9:30 and noon or 1:30 and 4, Monday through Friday. The number is (415) 652-8080. They'll try to answer your technical questions and provide more detailed product information than you may glean from our literature. It helps a lot if you think out your questions before you call, maybe make a few notes. It helps, too, if you have the manual and/or equipment handy to the phone. The easier it is for you to figure out exactly what we're talking about (and vice versa), the more help we can give in one phone call and the more phone calls we can handle. We'll also be happy to answer your questions if you drop us a letter.

There are two or three situations in which calling us isn't the fastest way to get help. If you have a problem with a **Processor** **Technology product which you purchased from a local dealer**, try your dealer first. They're all authorized to service the products they sell, precisely because they can give you faster and more personalized service than anybody can by mail. Our dealers all have diagnostic tapes available now, so if you need a PTC product tested it can be handled on the spot.

If you need a defective part replaced, just mail it to us with a note of explanation. There's nothing we can do about it over the phone anyway, so you might as well save yourself the time. (Naturally, it helps if your explanation is as clear and concise as possible. but we can't all he Hemingways.) If you need service that isn't covered by your warranty, please enclose a check for the \$20.00 minimum service charge. If the cost exceeds that (heaven forbid!) you will be notified before we proceed with the service.

If you do send in one of our products for repair, **please send only the defective board.** For obvious reasons, we cannot be responsible for the care and feeding of your monitors, keyboards or other products from other manufacturers.

MATCHMAKING-Software Division

Since most important programs are written for a particular system, modifications are sometimes in order to improve compatibility with your system. We'll be passing on more good ideas as we come across them; here are two dealing with MITS BASIC and the VDM.

MITS 12K Extended Disk and the VDM

If you don't have a 3P+S interface, you can still use the BASIC program in your VDM manual to link MITS 12K Extended Disc to your VDM driver. Simply change these six lines in the listing of VDM-1 to BASIC link appearing on page AV-18 of your VDM manual, and then run the program as explained there. Change the statements to read as follows: 0106 FOR Y=L TO L+419 0140 POKE G,195:POKE G+1,0:POKE G+2,P:POKE G+3,0 0160 DATA195,151,1001,195,156,1001,245,229,213,197 0162 DATA205,58,1000,193,209,195,147,1001,245,58 0232 DATA0,0,6,225,241,193,201,254,26,195,142 0234 DATA13,254,127,202,142,13,195,46,1000

MITS 8K or 12K BASIC and the VDM

If you have I/O ports that are assigned 20 and 21 instead of 0 and 1, you'll need to make five statement changes in the VDM-1 to BASIC link program in your VDM manual. The listing appears on pp. AV-17 to AV-19 of the manual, and should be modified as follows:

0078 IF A=219 AND 11=16 GOTO 92 0080 IF A=219 AND B=17 THEN I=K:GOTO 106 0152 DATA219,255,31,210,13,1000,219,16,230,2000 0224 DATA201,219,16,230,6000,201,58,140,1001,254 0228 DATA201,219,17,230,127,254,1,202,46,1000

MATCHMAKING-Hardware Division

When you're combining equipment from different manufacturers, optimum performance often depends on specific information about the idiosyncracies of their connections. Owner's manuals can't cover every permutation and combination, so think of this feature as a running supplement that gets down to cases.

3P+S and the OP 80A

We've had a lot of inquiries about using the OP-80A high speed manual punchpaper tape reader with the 3P+S, so we asked Oliver Audio Engineering for the straight scoop on interconnecting it with the 3P+S interface. Note that the OP-80A acknowledge line must be programmed for a negative true (ack) signal. We've also heard from many of our customers that the use of black tape and rolled tape (instead of fanfold) produces more accurate data transfer, using the OP-80.



Announcing the SOL USERS GROUP!

The Sol Users Group was recently organized by members of the Homebrew Computer Club in Palo Alto; SUG is not affiliated in any way with Processor Tech-nology. The purposes of SUG are to exchange software and other applications, and to create standards. If plenty of interest is shown, a Sol Newsletter will be published and sent to members.

If you own or have ordered a Sol, send your name, address, phone number and ideas to: Bill Burns

> 4190 Maybell Way Palo Alto, CA 94306 (Please don't call.)

PT 4KRA Memory Boards and the Motorola M6800

The following diagram should help you expand your M6800 system with minimum headaches. Note that the M6800 can drive 2 4KRA cards at reduced clock rates without bus drivers; however, full buffering is recommended.

CONNECTING PROCESSOR
TECHNOLOGY 4KRA MEMORY CARD
FOR USE IN MOTOROLA M6800
SYSTEMS



VDM-1 and the 6800 Microprocessor

The VDM-1 can live quite happily with your 6800 if a small amount of signal processing is performed externally.

Signals which may be directly used by the 6800 system are:

ADR0-ADR15	High-active address
DO0-DO7,	High-active data lines. They
DIO-DI7	may be connected in parallel
	to form a bi-directional data
	bus if necessary.

Signals which require conversion or re-interpretation:

i c mici pi ciuti	011.
PSYNC	Connect to high-active VMA
	(Valid Memory Address)
Φ2	Connect to $\Phi1$ clock
Pin 4 of IC 18	Break connection to Pin 3
(74LS132)	and connect to Pin 2

SINP, SOUT Connect to the highestorder address bits which are "1" when registers are addressed. (Bits 14 and 15 are suggested.) Presence of a "1" on either line will cause the address decoder to switch its comparison to to the status port address. The status port will therefore respond to any address whose top six bits are set by jumpers (see your VDM-1 manual), and whose bits 8 and 9 are zero. The loworder 8 bits are not decoded during status port response.

PWR	Externally generated signal consisting of VMA • Φ2 • R/W			
MWRITE	Inverted PWR			
PDBIN	Externally generated signal			

Externally generated signal consisting of VMA • $\Phi 2 \cdot R/W$

XRDY, PRDY Not used

Note that +8V, +16V and -16V are still required to operate the board, and that 6800 systems designed for all 5-volt operation may have to be augmented. Simple unregulated supplies will perform well, but care should be taken not to exceed +10V on the +8V line to avoid excessive dissipation in the +5V regulator.

What's the Best Monitor for your Sol or VDM-1?

That's one of the questions we're asked most frequently. We always recommend a black & white monitor designed for use with closed circuit TV systems or videotape recorders. Check your local Processor Technology dealer for good sources; they're up on that sort of thing.

If you want to use a regular black & white TV, try for a solid state model with an isolation transformer. In either case, remember that you don't have to spend a lot of money to get a decent picture, so let the rest of the family enjoy that big screen super color set.



Editor: Aram Attarian II

Publisher: Wible/Rampton Advertising, 727 15th Avenue, San Francisco, CA 94118 ACCESS is published approximately every six weeks. Subscription rate: \$4 per year, from Processor Technology Corp., 6800 Hollis St., Emeryville, CA 94608.

ACCESS Copyright © February 1977 by Processor Technology Corp. All Rights Reserved. Material in this publication may not be reproduced in any form without permission from Processor Technology Corp.

Interfacing a Keyboard and VDM with ALS-8, using the 3P+S.

To get your keyboard connection up and running, you'll need to connect 7 data lines, the keyboard strobe, and ground to the J2 connector of your 3P+S. Figure 1 shows the typical keyboard connections.

USING A KEYBOARD AND THE VDM WITH THE ALS-8

KEYBOARD INTERFACE

TYPICAL KEYBOARD CONNECTIONS



The keyboard strobe. This circuit tells the processor when a key has been pressed. You'll need to install an additional IC (preferably in a 16 pin DIP socket) in the unused IC pad in the lower right corner of your 3P+S, and connect it to pin 7 of J2. It's used to condition the strobe. Use a 74LS109 or 74109. (Figure 2)



To wire the 74109:

- 1) Connect pins 2 and 3 to ground.
- 2) Pins 16 (VCC) and 8 (GND) are already
- connected on board. 3) Connect pin 1 to pin 1 of IC 15 to provide
- a pull-up connection.
- 4) Connect pin 5 to pin 11 of IC 15 to reset the flip-flop when data has been accepted.
- 5) Connect pin 7 to a point on 3P+S leading to J2 pin 12. This point will go low when the strobe occurs.
- 6) Connect pin 4 to a point on 3P+S leading to J2 pin 7, the point where the strobe from the keyboard will be connected.

The data lines. Either seven or eight data lines are used to transmit the ASCII code for the key being pressed to the computer. They're connected to the B port inputs of the 3P+S in the following manner:

Keyboard	3P+S J2	
<u>Signal</u>	Connector	<u>Data Bit</u>
Bit 1	Z	D0
Bit 2	Y	D1
Bit 3	Х	D2
Bit 4	W	D3
Bit 5	V	D4
Bit 6	U	D5
Bit 7	Т	D6
Bit 8*	S	D7

*Pin J2-S is left unconnected if your keyboard doesn't have an eighth bit.

Ground, +5V, -12V. These provide power for the circuitry of the keyboard. Ground is simply connected to pin 12 of J2. +5V and -12V (regulated) should be provided by a separate power supply. Some KYBDS only require +5V Figure 3 shows an example of one, assembled on a small piece of perf board and attached to the keyboard or main frame. Jumper it to the keyboard connector.

ASSEMBLE ON SMALL PIECE OF PERF BOARD AND ATTACH TO KYBD OR MAINFRAME AND JUMPER TO KYBD CONNECTOR



Jumper areas. Only two are important for the keyboard interface: Area A must be jumpered for address 0 (i.e., all to ground), and Area B must be jumpered from left to center, to set port A at 0 (status) and port B at 1 (data) to correspond to Processor Technology software.

Testing the interface with ALS-8.

1) Turn on the computer and examine location E000H. When you hit Run, the address lights should look like this:

A15	A14	A13	A12	All	A10	Α9	A8
•	•	0	•	0	0	0	•

• = Light on o = Light out

 Be sure the keyboard is sending upper case characters. Type EXEC E024, then hit the carriage return key. Address light A13 should blink instantaneously at this point.

- 3) Type IODR/SYSIO/0 FE77, then hit the return key. Again, address light A13 should blink.
- 4) Now hit Space, Space, Return; wait a moment and then hit Control Z. The screen should now be blank.
- 5) Hit Control S. The word SPEED? should now appear in the lower left corner of the screen. If it doesn't, you may have to adjust the horizontal and vertical controls on the VDM to get this message onto the visible portion of the screen.
- 6) Type 1; you should get an automatic carriage return, then type DUMP 0 FFFF and hit Return. Memory will now dump on the VDM display. You can hit the Escape key to stop the dump.
- Now you're ready to start programming with the ALS-8! Try some of the examples from the ALS-8 manual to familiarize yourself with its operation.

Hoare's Law of Large Programs Inside every large program is a small program struggling to get out.

Good News for our Customers in Europe: VDM-1 and the European 50 Hz Standard

The European television standard maintains the same horizontal rate as the U.S. (15,750 Hz), but it defines a raster of 625 lines at a field rate of 50 Hz. The effect is to increase the number of scan lines on the screen.

It's quite easy to modify your VDM-1 to work on this standard. Simply disconnect pin 5 of IC 8 from pin 6, and reconnect it to ground (pin 4). This increases the modulus of the counter to 8 in the VDISP time, resulting in 4 extra character lines (52 scan lines) on the display. The total is now 312 scan lines per field, the equivalent of 624 per frame.

The field rate should be close enough to 50 Hz to reduce any swim effects to less than 0.1 Hz. You may have some trouble centering the display in the frame, because the standoff time to VSYNC from the bottom of the display is still on the 60 Hz standard. If the effect is objectionable, increase Resistor R 34 in series with the VPOS control.

Rule of Accuracy

When working toward the solution of a problem it always helps you to know the answer. Sattinger's Law

Newett Awl's Choo Choo Train Or, Idle Fantasies on a VDM Screen

						(Once upon	a time, in a curious little
						place, th	nere was a	Sol system and a program-
						mer sitti	ing around	with nothing to do. So
						Newett	Awl decide	ad to tell his computer a
0000				0000		1 14		VDM 1 halved him seet has
0000				0010		beatime	story, and	VDM-1 helped him out by
0000				0020		drawing	the pictur	es.
0000				0030		-	Trv it on vo	our system. Ol' Uncle Sol
0000				0040		makes a	great haby	vsitter
0000				0050		makes a	i gicai baby	sitter.
0000				0060				
0000	7 17			0070			7	ONCE LIDON & ETME
0000	AF A7	CB		0071	IRAIN		A OCSH	IN A CUPINIA LIME,
0001	21	84	01	0080		TXT	H.SHED	THERE WAS
0006	01	00	08	0090		XSI	в,2048	A TINY TRAIN
0009	36	20		0100	EMPTY	MVI	М,20Н	AND IT STAYED IN A
000B	23			0110		INX	H	TINY SHED
000C	0B			0120		DCX	В	THAT WAS ALL EMPTY
000D	AF.			0130		XRA	A	
0005	A0 C2	09	0.0	0140		ARA JNZ	d FMDTV	
0012	21	57 F7	02	0160		LXI	H. CLOUD	EXCEPT FOR A HUGE
0015	11	00	00	0170		LXI	D,0	
0018	06	0F		0180		MVI	в,15 в	
001A	CD	50	00	0190		CALL	SMO1 I	
001D	11	2B	00	0200		LXI	D,43 1	- -
0020	06		0.0	0210		MVL	B,13 SMO1	
0022	11	30	00	0220		LXT	D 48	W
0028	06	09	00	0240		MVI	B,9	Y
002A	CD	50	00	0250		CALL	SM01	-
002D	11	34	00	0260		LXI	D,52	С
0030	06	04		0270		MVI	в,4	L
0032	CD	50	00	0280		CALL	SMO1	0
0035	11 06	3B 02	00	0290		LXI	D,59	U
0038 0038	CD	50	0.0	0300		CALT.	SMO1	D
003D	11	3C	00	0320		LXI	D,60	0
0040	06	01		0330		MVI	В,1	F
0042	CD	50	00	0340		CALL	SMO1	_
0045	11 06	3년 01	00	0350		LXI	D,62	S
0040 004a	CD	50	0.0	0370		CALL	SMO1	N O
004D	C3	59	00	0380		JMP	ENGIN	ĸ
0050				0390	•			E
0050	19	C-		0400	SMO1	DAD	D	0011011
0051	36	6F.		0410	SMO2	MVL	M, SMOKE	COUGH
0053	23 05			0420		DCB	п В	COLICH
0055	C2	51	00	0440		JNZ	SMO2	
0058	C9			0450		RET	. C(OMING OUT OF THE STACK OF
0059	0.1			0460	•			
0059	21	A6	04	0470	ENGINE	LXI	H, SHED+32	22H A TINY LOCOMOTIVE
005C	21	то д9	04	0480		T.XT	H SHED+3	25H AND A LITTLE BELL
0061	36	07	01	0500		MVI	M,07H	
0063	21	AB	04	0510		LXI	H,SHED+32	27HAND A TINY DOME
0066	36	6E		0520		MVI	М,бЕН	
0068	2A	66	01	0530		LHLD	CAB1 .	AND A CAB
006B	22 27	AE 68	04	0540		SHLD	CAR2	SO THAT YOU COULD
0071	22	в0	04	0560		SHLD	SHED+32CI	H SEE INTO WHERE THE
0074	2A	6A	01	0570		LHLD	CAB3 .	. ENGINEER AND THE
0077	22	в2	04	0580		SHLD	SHED+32E	H FIREMAN SAT.
007A	2A	6C	01	0590		LHLD	BOI1 .	. IT HAD A BEAUTIFUL
007D	22 27	Ľ5 ۲	04 01	0600		SHLD	SHED+3611	H PULISHED
0080	∠A 2.2	E7	04	0620		SHLD	SHED+363	H BOILER
0086	2A	70	01	0630		LHLD	BOI3 .	. WITH
0089	22	E9	04	0640		SHLD	SHED+365	H THE NUMBER
008C	22	EB	04	0650		SHLD	SHED+367	H "99" ON THE
008F,	∠A 20	ע ריק	01 04	0650 0670		СЦНЦ СТНГО	SHEDT3eoi	N SIDE OF THE CAR
0095	2A	74	01	0680		LHLD	BOI5 .	. BUT YOU COULDN'T
0098	22	EF	04	0690		SHLD	SHED+36B	H SEE EITHER THE
	0 7	76	01	0700		T.HT.D	BOIG	FNGINFFR OR THF

(continued on page 11)

Ups and Downs or How to Type in Upper Case Only without Shifting

If you have a keyboard with both upper and lower case operation, the frequent shifts are a pain when you're entering alphanumeric data. Here's a simple cure that sets data bit D5 low when a lower case alphabetic character is output from the keyboard, thus presenting it to the computer as upper case.

You'll need two chips, either a 7404 and a 7408 or a 74LS04 and a 74LS08. (Either pair works fine.) Install them on a small piece of perf board as shown in the diagram, and tie all unused inputs to +5V.

The pins you'll be using on the 04 chip are 3, 5, 9, 11, and 13; on the 08 chip use 4, 5, 9, 10, 12, and 13.



FLASH ~~ VDM Access Flicker Eliminated

That blasted flicker you get whenever the VDM memory is being frequently accessed occurs because the screen is blanked for a very short time whenever the processor reads or writes to memory. You can eliminate it by modifying the timing system so than VDM memory is accessed only when the beam of the picture tube is off the visible portion of the screen. The VDM has a timing signal that indicates this condition, and you can use it to synchronize access to display memory.

Connect a jumper wire from pin 13 of IC 39 to pin 5 of IC 39. This hooks up an unused section of IC 39 to DI bit 1. Connect a wire from IC 39 pin 14 to IC 15 pin 9. (This allows programs which access the VDM memory to use the timing signal.) Now when C8 is input, bit 1 will be low whenever the display memory can be accessed.

We'll explore the implications further in the next issue, and provide an example program or two.

Murphy's Law of Thermodynamic Things get worse under pressure. Lowery's Law

If it jams—force it. If it breaks, it needed replacing anyway.



Application News Ham Computer Based on Sol Terminal System

SYSTEM 4000 ham computer was developed by Curtis Electro Devices, Inc., Box 4090, Mountain View, Ca. 94040. The company makes ham keyers and an industrial line of PROM programmers. We asked the president, John G. Curtis, to comment on working with the Sol system; he did all of the hardware and software work himself, developing programs for the Sol Terminal on ALTAIR with ALS-8, VDM 1, 40K of RAM, 3P+ 1, Tarbell cassette system, Bytesaver, ASR-33 TTY, and Motorola video monitor. Jack's report was glowing:

"I had no previous experience or trainin software development. It was all learned on the fly from books and practical experience. (Try, try and try again!!!) . . . Not too much assistance was required. The ALS-8 is easy to use and the Simulator program is absolutely essential. The Editor is also absolutely necessary. (Everything is necessary!!)...

"The Sol went together with a minimum of effort and trouble even though it was one of the first units actually put on the line. There were things which didn't work but with the help of PTC (especially Bob Marsh and Aram) things were put in order in a hurry.

hurry. "In my opinion, the Sol terminal is the most ideal small computer system available today (certainly at that price) and every needed feature is there. For our purposes it was absolutely ideal from every standpoint. We are now able to get the computer into the hamshack on a commercial basis. This has been predicted for sometime, now it has happened."

System 4000 Ham Computer

The SYSTEM 4000 is a full scale standard desk top minicomputer specially equipped with firmware programs and interfaces for the amateur radio operator. Being a stand-alone computer, it can also run business, educational, scientific or games programs.

The SYSTEM 40000 is derived from the Sol terminal Computer and can take advantage of Processor Technology's programs and accessories. System 4000 uses the standard S-100 bus for plug-in accessory boards, and the owner can add or exchange PROM integrated circuits to update the system. Features:

Morse reader, capable of receiving code at speeds of 6-250 WPM (or higher). CRT or TTY output, selectable for upper and lower case.

Paddle keyer, with dot= and dash memory, iambic and full self completion. CRT or TTY printout.

Keyboard keyer, sends Morse from keyboard. CRT or TTY output.

ASCII terminal, half or full duplex.

Complete details can be obtained by writing to: Curtis Electro Devices, Inc., P O. Box 4090, Mountain View, Ca. 94040.

Murphy's Third Law

In any field of scientific endeavor, anything that can go wrong will go wrong.

Sevarenid's Law

The chief cause of problems is solutions. Brooks's Law

Adding manpower to a late software project makes it later.

O'Tooles's Commentary on Murphy's Laws Murphy was an optimist.



A major purpose of this newsletter is helping you stomp out the bugs that inev-itably occur even in the best of systems. Bug Squad will be a regular feature. We'll tackle the problems we've encountered most frequently, and you're also invited to send in any problems that have been bugging you. Send solutions, too, if you've found them. Share the wealth.

For starters, here's how to fix a few of the bugs that crept into early Sol systems:

Memory Protect/Unprotect Lines *The bug*: Erratic behavior when early Sol-PC, Sol-10, and Sol-20 units are used with S-100 bus compatible memory modules. *The squasher*: On early Sol circuit boards, the protect (pin 70) and unprotect (pin 20) lines are floating. Simply ground bus line 70 on the Sol PCB itself to disable the memory protect signal.

Current Loop Source

The bug: R23, a 470 ohm 1/2w resistor, is incorrectly tied to +5 on early Sol PC boards, producing less than a full 20mA current to teletype connections.

The squasher: Reconnect R23 to + 12V which is only 1/4" away. Check to make sure that R29 remains connected to +5V. We suggest that the 20mA current loop connections (such as for ASR33) be made as follows:



Accidental Avalanche

The bug: On some Sol Power Supply REG Boards, the SCR1 (MCR 106-2) in the +5 volt regulated supply will intermittently go into the avalanche mode. This draws sufficient current to drop the +5 volt to about +1V This modification will prevent the accidental triggering, but will retain the

overvoltage protection of the crowbar circuit.

The squasher: Make the modification here by adding the components shown and cutting appropriate traces (as shown).



Law of Selective Gravity (The Buttered Side Down Law)

An object will fall so as to do the most damage.

Where the %×&*# is Emeryville?

Emeryville, Ca., is located at the foot of the Oakland Bay Bridge, squeezed between Oakland and Berkeley. Neither wishes to claim it. Emeryville consists largely of mudflats and factories, the principle landmarks being a highrise apartment complex called Watergate (no relation), a shoreline assemblage of driftwood sculptures, and Processor Technology. Before the rise of Processor Technology, the main industry was legal gambling clubs. We, however, plan to put Emeryville on the map.



Consol Source Listing

For those of you who haven't seen it yet, here's the source list for the minimum Sol operating System, CONSOL. It provides all necessary display routines, along with standardized calling points for input/output operations.

-== CONSOL ==-COPYRIGHT 1976



C004	0052	*		= SoL SY	STEM I/O ROUTINES =
C004	0054	* THE	FOLLOWING	CODE IS ST	ANDARDIZED FOR ALL SOL SYSTEM
C004	0056	* OUTP	UT OPERAT	IONS. CONS	OL DOES NOT HAVE PROVISION FOR BECAUSE OF SPACE LIMITATIONS
C004	0058	*	1060 170	OFERALIONS	BECAUSE OF SPACE HIMITATIONS.
C004 C004	0059	*			
C004 C004	0062	*	JUMP	TABLE INPU	T/OUTPUT ROUTINES
C004 C004	0063	* THIS	ROUTINE	OUTPUTS THE	CHARACTER IN REGISTER 'B' TO
C004 C004	0065	* THE OU * REGIST	TPUT DEVI ER. THE	CE POINTED DEVICES ARE	DEFINED AS FOLLOWS:
C004 C004	0067	*	0 - VDM S	CREEN	
C004 C004	0069 0070	*	1 - SERIA 2 - PARAL	L OUTPUT PO LEL OUTPUT	RT PORT (NOT AVAILABLE ON CONSOL)
C004 C004	0071 0072	*	3 - ERROR	HANDLER	
C004 C004	0073 0074	* ENTR *	Y AT: SO AO	UT SELECTS UT SELECTS	CURRENT OUTPUT DEVICE DEVICE IN REGISTER 'A'
C004 C004 3A 03 C8	0075 0076	* SOUT	LDA	OPORT	GET PORT NUMBER FROM MEMORY LOCATION
C007 E6 03 C009 E5	0077	AOUT	ANI PUSH	3 H	KEEP IT IN CONTROL WE'LL RESTORE IT LATER
COUA 21 84 CI COOD 07	0079		RLC	H,OTAB	COMPUTE ADDRESS
COOF 6F	0081		ADD MOV	L L,A	WE HAVE IT
C010 C3 /1 C0 C013	0083	*	JMP	DISPT	GO TO HL
C013 C013	0085	*	DOUTTINE	TNDURG & GU	ADAGED TO DECLEMED IAL EDOM
C013 C013	0087	* THE CU	ROUTINE RRENT INP	UT DEVICE P	OINTED TO BY THE CURRENT INPUT
C013 C013	0089	* ENTER	REGISTER		
C013 C013	0091	* ENIR	0 VEV	ARE DEFINED	•
C013 C013	0094	*	1 - SER	IAL INPUT	(NOT AVAILABLE ON CONSOL)
C013 C013	0095	*	3 - ERR	OR HANDLER	(NOT AVAILABLE ON CONSOL)
C013 3A 04 C8	0098	SINP	LDA	IPORT	GET PORT NUMBER FROM MEMORY LOCATION
C018 E5	0100	AINF	PUSH	н н ц ттрр	SAVE H&L
C01C 07 C01D 85	0102		RLC	, ± 1 MD T.	THE MATH
C01E 6F C01E C3 71 C0	0104		MOV	L,A	DONE
C022 C022	0105	*	OMP	DISFI	WE HAVE THE ADDRESS GO TO HEAVEN
C022 C022	0108	*	KEYBOARD	INPUT STATU	S CHECK
C022 C022	0110	* THIS * WITH	ROUTINE T	ESTS THE KE BITS SET	YBOARD STATUS AND RETURNS
C022 C022 DB FA	0112	* KSTAT	TN	STAPT	GET STATUS WORD
C024 E6 01 C026 C9	0114	ROINI	ANI	KDR	TEST KEYBOARD BIT FLAGS ARE SET
C027 C027	0116	*	ICH I	·	
C027 C027	0118	*	KEYBOARD	DATA INPUT	
C027	0120	* THIS	ROUTINE,	ENTRY AT KR	EAD, GETS THE DATA FROM THE
C027	0121	* KEYBO	ARD. ON	RETURN THE	CHARACTER IS IN REGISTER 'A'.
C027 C027 C027 C027 E1	0121 0122 0123	* KEYBO * KREA1	ARD. ON POP	RETURN THE	CHARACTER IS IN REGISTER 'A'.
C027 C027 C027 E1 C028 CD 22 C0 C02B C2 28 C0	0121 0122 0123 0124 0125	* KEYBO * KREA1 KREAD	ARD. ON POP CALL JNZ	RETURN THE H KSTAT KREAD	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT
C027 C027 C027 E1 C028 CD 22 C0 C02B C2 28 C0 C02E DB FC C030 C9	0121 0122 0123 0124 0125 0126 0127	* KEYBO * KREA1 KREAD	ARD. ON POP CALL JNZ IN RET	RETURN THE H KSTAT KREAD KDATA	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT
C027 C027 C027 E1 C028 CD 22 C0 C028 C2 28 C0 C02E D8 FC C030 C9 C031 C031	0121 0122 0123 0124 0125 0126 0127 0128 0129	* KEYBO * KREA1 KREAD * *	ARD. ON POP CALL JNZ IN RET	RETURN THE H KSTAT KREAD KDATA •	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT
C027 C027 C027 E1 C028 CD 22 C0 C028 CD 22 C0 C02E DB FC C030 C9 C031 C031 C031 C031	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131	* KEYBO * KREAL KREAD * * * SERI	ARD. ON POP CALL JNZ IN RET AL INPUT	RETURN THE H KSTAT KREAD KDATA · STATUS CHEC	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K
C027 C027 E1 C028 CD 22 C0 C028 CD 22 C0 C028 CD 22 C0 C020 C2 28 C0 C021 DB FC C031 C031 C031 C031 C031 DB F8 C033 E6 40	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133	* KREA1 KREA1 KREAD * * * SERI * SSTAT	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY
C027 C027 C027 C028 CD 22 C0 C028 CD 22 C0 C028 C2 28 C0 C022 DB FC C030 C9 C031 C031 C031 C031 C031 C031 DB F8 C033 E6 40 C035 C9 C036 C036 C036 C036 C036 C036 C036 C036	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133 0134 0135	* KEYBO * KREA1 KREA1 KREAD * SERI * SERI *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET	RETURN THE H KSTAT KREAD KDATA · STATUS CHEC SERST SDR ·	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET
C027 C027 C027 C028 CD 22 C0 C028 CD 22 C0 C028 C2 28 C0 C026 C2 28 C0 C030 C9 C031 C031 C031 C031 C031 C031 DB F8 C033 E6 40 C035 C9 C036 C036 C036 C036	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133 0134 0136 0137	* KEYBO * KREA1 KREA1 KREAD * SERI * SERI * SERI * SER	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET
C027 C027 C027 C028 CD 22 C028 CD 22 C02E DB FC C030 C031 C031 C031 C031 C031 C031 C03	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139	* KEYBO * KREA1 KREA1 KREAD * * SSTAT * * SSTAT * * SSTAT	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST JDR INPUT H	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY
C027 C027 C027 C028 C029 C020 C029 C029 C020 C029 C029 C029 C020 C029 C021 C025 C026 C027	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139 0140 0141	* KEYBO * KREA1 KREA1 KREAD * SERI * SERI * SERA SREA1 SREAD	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ	RETURN THE H KSTAT KREAD KDATA · STATUS CHEC SERST SDR · INPUT H SSTAT SREAD	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT
C027 C027 C027 C028 C028 C028 C028 C022 C028 C022 C028	0121 0122 0123 0124 0125 0126 0127 0128 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139 0140 0141 0142 0143	* KEYBO * KREA1 KREA1 KREAD * * SSTAT * * SSTAT * * SSTAT * * SSTAT	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET	RETURN THE H KSTAT KREAD KDATA · STATUS CHEC SERST SDR · INPUT H SSTAT SREAD SDATA ·	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT GET DATA BYTE WE HAVE IT
C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139 0140 0141 0142 0143 0144	* KEYBO * KREAI KREAI KREAD * SERI * SERI * SERA SREA1 SREAD	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SREAD SDATA	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT GET DATA BYTE WE HAVE IT
C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029	0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0132 0133 0133 0134 0135 0136 0137 0138 0139 0140 0141 0142 0143 0144 0145 0146	* KEYBO * KREAI KREAI * * * * * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SREAD SDATA OUTPUT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT GET DATA BYTE WE HAVE IT
C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C030 C031 C035 C036 C036 C036 C036 C036 C036 C036 C036 C037 C031 C031 C031 C031 C031 C031 C031 C035 C036 C036 C036 C036 C036 C036 C036 C036 C037 C040	0121 0122 0123 0124 0125 0126 0128 0129 0130 0131 0132 0133 0134 0133 0134 0133 0134 0133 0134 0133 0134 0143 0144 0145 0146 0147	* KEYBO * KREA1 KREA1 KREAD * * SSTAT * * * SSTAT * * * SREA1 SREA1 SREA1 SREA1 SREA1 SREA1 * * * * * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IN RET	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR · INPUT H SSTAT SREAD SDATA · OUTPUT H SERST	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C029 C020 C029 C020 C021 C022 C026 C027 C021 C027 C021 C027	0121 0122 0122 0125 0126 0125 0126 0125 0126 0127 0128 0127 0128 0127 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139 0130 0131 0132 0133 0134 0135 0146 0148 0149 0148 0149 0148 0149 0148 0149 0148 0148 0149 0148 0149 0148 0149 0148 0149 0148 0149 0148 0149 0148 0149 0148 0149 0148 0149 0148 0149 0148 0148 0148 0148 0148 0148 0148 0148	* KEYBO * KREAI KREAI KREAD * SSTAT * SSTAT * SSTAT * SSREAI SREAI SREAI SREAI SREAI SREAT SSREAT	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA	RETURN THE H KSTAT KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SREAD SDATA OUTPUT H SERST SDROT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS
C027 C027 C027 C027 C028 C028 C028 C022 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C020 C020 C021 C036 C036 C036 C036 C036 C036 C037 C037 C031 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C036 C036 C036 C036 C036 C036 C036 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C037 C036 C036 C036 C036 C036 C036 C036 C037 C040 C040 C040 C040 C040 C041 C047	0122 0122 0123 0124 0125 0126 0125 0126 0127 0128 0127 0128 0127 0130 0130 0130 0130 0130 0130 0135 0136 0137 0138 0136 0137 0138 0137 0138 0137 0140 0141 0142 0143 0144 0145 0151	* KEYBO * KREA1 KREA1 KREAD * * SERI * SSTAT * SSTAT * SERA1 SREAD * * SER * SERAT SREAD	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SSTAT SSTAT SOUTPUT H SERST SDR A,B SDATA	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT
C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C020 C020 C020 C020 C021 C026 C027 C02 C027 C02 C027 C02 C027 C02	0121 0122 0123 0124 0125 0126 0125 0126 0127 0128 0127 0128 0127 0130 0130 0131 0132 0133 0134 0135 0136 0137 0138 0136 0137 0138 0139 0140 0141 0142 0144 0145 0146 0147 0151 0151	* KEYBO * KREAI KREAI KREAD * * SSTAT * * SSTAT * * SREAI SREAD * * SREAD * SREAD	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDATA SDATA A, B SDATA	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C030 C031 C036 C036 C036 C036 C036 C036 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C036 C036 C036 C036 C037 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C036 C036 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 S F9 C040 C	0122 0122 0123 0124 0125 0126 0126 0127 0128 0126 0127 0128 0130 0130 0130 0131 0132 0136 0136 0137 0138 0136 0141 0148 0148 0148 0148 0148 0148 0148	* KEYBO * KREA1 KREA1 KREA1 * * SSTAT * * * SSTAT * * * SREA1 SREA1 SREA1 SREA1 SREA1 SREA1 * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP CALL JZ IN RET IAL DATA	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SDATA OUTPUT H SERST SDR A,B SDATA	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR INPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C030 C031 C036 C036 C036 C036 C036 C036 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C036 C036 C036 C036 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C036 C036 C036 C036 C036 C037 C0 31 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C036 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 C037 C0 31 C0 C040 C040 C040 C040 C040 C040 C040 C040 C040 C047 C0 SF9 C048 C	0122 0123 0122 0123 0124 0125 0126 0126 0127 0128 0127 0128 0130 0131 0132 0130 0131 0132 0133 0134 0135 0141 0142 0143 0144 0145 0153 0154 0155 0156	* KEYBO * KREA1 KREA1 KREA1 * * SSTAT * * * SSTAT * * * SREA1 SREA1 SREA1 SREA1 SREA1 SREA1 SREA1 * * * * * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR · INPUT H SSTAT SREAD SDATA · OUTPUT A,B SDATA ·	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE
C027 C027 C027 C028 C029 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C035 C036 C036 C036 C036 C036 C036 C036 C036 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C036 C036 C036 C036 C036 C036 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C036 C036 C036 C036 C036 C036 C037 CD 31 C0 C040	0122 0122 0123 0124 0125 0126 0126 0126 0127 0128 0127 0128 0130 0132 0131 0132 0133 0134 0132 0133 0136 0137 0138 0139 0140 0149 0150 0155 0156 0155 0155 0155 0155 0155	* KEYBO * KREA1 KREA1 KREAD * * SSTAT * * SSTAT * * SSTAT * * SSREA1 SREA1 SREA1 SREA1 SREA1 SREA1 * * * * * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSEAD SDATA OUTPUT H SDROT A, B SDATA VIDEO DISPL	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES
C027 C027 C027 C028 C029 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C033 C036 C036 C036 C036 C036 C036 C036 C037 C031 C031 C037 C031 C037 C031 C037 C037 C037 C037 C037 C037 C037 C037 C039 C036 C036 C036 C036 C036 C036 C037 C040 C040 C040 C040 C040 C040 C040 C047 78 C048 C0	01221 01222 01233 0124 01255 01266 01267 0128 0129 0130 0132 0130 0132 0130 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0141 0142 0143 0145 0145 0145 0145 0145 0145 0145 0145	* KEYBO * KREA1 KREA1 KREA1 * * SSTAT * * SSTAT * * SSTAT * * * SSREA1 SREA1 SREA1 SREA1 SREA1 SREA1 * * * * * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA	RETURN THE H KSTAT KSTAT KTATUS CHEC SERST SDR INPUT H SSTAT STATUS STAT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL
C027 C027 C027 C027 C028 C028 C028 C022 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C031 C032 C036 C036 C036 C036 C036 C036 C037 C040 C040 C040 C040 C048	01221 01222 01233 0124 01255 01266 01267 0128 0129 0130 0131 0132 0131 0132 0131 0132 0133 0134 0135 0136 0137 0138 0139 0140 0141 0142 0143 0144 0145 0146 0147 0148 0145 0146 0147 0155 0155 0155 0155 0155 0155 0155 015	* KEYBO * KREAL KREAL KREAL * SERI * SERI * SERAL SREAL SREAL SREAL SREAL SREAT SREAT * SER *	ARD. ON POP CALL JNZ IN RET AL INPUT IAL INPUT IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JC IN RET IAL DATA POP CALL JC IN RET IAL DATA POP CALL JC IN RET IAL DATA POP CALL JC IN RET RET RET	RETURN THE H KSTAT KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SREAD SDATA OUTPUT H SERST A,B SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL PERT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN EPS ABE UNALTERED
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C032 C036 C036 C036 C036 C036 C036 C036 C036 C037 CD 31 C0 C037 CD 31 C0 C037 CD 31 C0 C037 CD 31 C0 C036 C036 C036 C036 C036 C036 C036 C037 CD 31 C0 C037 CD 31 C0 C037 CD 31 C0 C037 CD 31 C0 C037 CD 31 C0 C036 C036 C036 C036 C036 C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C036 C036 C036 C036 C036 C036 C036 C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C037 CD 31 CO C040 C040 C040 C040 C040 C040 C040 C048 C04	0122 0123 0122 0123 0124 0125 0126 0126 0127 0128 0126 0127 0128 0130 0130 0131 0132 0130 0131 0132 0133 0134 0135 0136 0137 0138 0136 0141 0142 0155 0146 0155 0156 0156 0156 0157 0158 0156 0156 0157 0158 0156 0156 0157 0158 0156 0156 0157 0158 0156 0156 0156 0157 0158 0156 0156 0156 0156 0156 0156 0156 0156	* KEYBO * KREA1 KREA1 KREA1 * * SSTAT * * SSTAT * * * * * SREA1 SREA1 SREA1 SREA1 SREA1 * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET RET IAL DATA POP IN RET RET IAL DATA POP COUTINES COUTINES ICONSOLO E AND CONSOLO IN RET	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SREAD SDATA OUTPUT H SERST SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST VUDEO DISPL	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR INPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS BOUTINE IS A MINIMUM
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C031 C032 C036 C036 C036 C036 C036 C036 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C036 C036 C036 C036 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C036 C036 C036 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C037 C0 31 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 S F9 C040 C040 C040 C040 C040 C040 C040 C040 C048	0122 0123 0122 0123 0124 0125 0126 0126 0127 0128 0120 0130 0131 0132 0130 0131 0132 0133 0134 0135 0136 0141 0142 0143 0144 0145 0146 0147 0158 0150 0151 0158 0156 0157 0158 0159 0150 0157 0158 0155 0156 0157 0158 0159 0150 0157 0158 0157 0158 0159 0150 0157 0158 0159 0150 0151 0158 0159 0150 0151 0151 0151 0151 0151 0151	* KEYBO * KREA1 KREA1 KREAD * * SSTAT * * SSTAT * * * * * SREA1 SREA1 SREA1 SREA1 SREA1 SREA1 * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP CALL JNC MOV OUT RET RET ION RET	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR · INPUT H SSTAT SREAD SDATA · OUTPUT H SERST SDATA A,B SDATA · VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR IMPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY:
C027 C027 C027 C027 C028 C030 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C035 C036 C036 C036 C036 C036 C036 C036 C036 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C036 C036 C036 C036 C036 C036 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C037 CD 31 C036 C036 C036 C036 C036 C036 C037 CD 31 CO C040 C048	0122 0122 0123 0124 0125 0126 0126 0126 0127 0128 0127 0128 0130 0132 0131 0132 0131 0132 0133 0136 0137 0138 0136 0137 0138 0141 0142 0143 0149 0150 0155 0156 0157 0158 0155 0156 0157 0158 0155 0156 0157 0158 0155 0156 0157 0158 0155 0156 0157 0158 0155 0156 0157 0158 0155 0156 0157 0158 0157 0158 0159 0150 0157 0158 0159 0150 0157 0158 0159 0150 0157 0158 0159 0150 0157 0158 0159 0150 0157 0158 0155 0156 0157 0158 0155 0156 0157 0158 0155 0156 0157 0158 0157 0158 0159 0150 0157 0158 0157 0158 0157 0158 0156 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0159 0150 0150 0157 0158 0159 0150 0150 0150 0150 0150 0150 0150	* KEYBO * KREA1 KREA1 KREA1 * * SSTAT * * SSTAT * * SSTAT * * SSTAT * * SSTAT * * * SSTAT * * SSTAT * * * * SSTAT * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA IN RET IAL DATA IN RET IAL DATA IN RET IAL DATA IN RET IAL DATA IN RET IAL DATA IN IN IAL DATA IN IN RET IAL DATA IN IN IN IN IN IN IN IN IN IN	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SREAD SDATA OUTPUT H SERST OUTPUT H SDROT A,B SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE AN KETLEBO	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C033 C036 C036 C036 C036 C036 C036 C036 C037 C037 C031 C037 C040 C040 C040 C040 C048	01221 01222 01233 01244 01255 01266 01267 01286 01277 01288 01297 01300 01311 0132 01301 0132 01331 0134 0132 01331 01340 01321 01331 01340 01321 01331 0144 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0141 01450 0157 01580 0159 01500 0151 0150 0150 0151 0150 0170 010000000000	* KEYBO * KREA1 KREA1 KREA1 * * SSTAT * * * SSTAT * * SSTAT * * * * SSTAT * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RAL JNC MOV OUT RET IONS. O TER B AND E 'CONSOL MENTATION I COL	RETURN THE H KSTAT KSTAT KTATUS CHEC SERST SDR INPUT H SSTAT STATUS CHEC SERST STATUS CHEC SERST OUTPUT H SERST OUTPUT H SERST VIDEO DISPL ALLOW FOR N ENTRY, TH ALLOW FOR N ENTRY ALLOW FOR N ENTRY AL	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C032 C036 C036 C036 C036 C036 C036 C036 C037 C040 C040 C040 C040 C040 C040 C040 C040 C040 C048	01221 01222 01233 0124 01255 01266 01267 0128 0129 0130 0131 0132 0131 0132 0133 0134 0133 0134 0135 0136 0137 0138 0139 0141 0142 0133 0144 0145 0147 0148 0147 0148 0147 0152 0153 0154 0155 0156 0157 0158 0159 0150 0151 0152 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0151 0155 0156 0157 0158 0157 0158 0159 0150 0151 0151 0155 0156 0157 0158 0157 0158 0159 0150 0151 0155 0156 0157 0158 0157 0158 0159 0150 0151 0157 0158 0159 0150 0151 0155 0157 0158 0159 0150 0151 0158 0157 0158 0159 0150 0151 0158 0157 0158 0159 0150 0151 0158 0157 0158 0159 0150 0151 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0177 0158 0177 0158 0177 0158 0177 0158 0177 0158 0177 0178 0177 0178 0177 0178 0177 0177	* KEYBO * KREAL KREAL KREAL * SERI * SERI * SERI * SERAL SREAL SREAL SREAL SREAL SREAL * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * * * SER * SER * SEROT SDROT * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA IN IN RET IAL DATA IN IN RET IAL DATA IN IN IN RET IAL DATA IN IN RET IAL DATA IN IN IN RET IAL DATA IN IN RET IAL DATA IN IN IN RET IAL DATA IN IN IN IN IN IN IN IN IN IN	RETURN THE H KSTAT KSTAT KSTATUS CHEC SERST SDR INPUT H SSTAT SSTAT STATUS CHEC SERST OUTPUT H SERST SDROT A,B SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE AN KETTLEBO OF LEGE STATIO	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RES.
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C032 C036 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C030 C0 C0 C0 C0 C0 C0 C0 C0 C0 C	0122 0123 0122 0123 0124 0125 0126 0126 0127 0128 0130 0130 0131 0132 0130 0131 0132 0133 0134 0135 0136 0141 0142 0143 0144 0145 0146 0147 0151 0152 0153 0156 0157 0158 0156 0157 0158 0159 0150 0151 0152 0153 0156 0157 0158 0156 0157 0158 0159 0150 0151 0152 0153 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0156 0157 0158 0157 0158 0156 0157 0158 0157 0158 0159 0150 0151 0157 0158 0156 0157 0177 0158 0156 0177 0158 0156 0157 0177 0158 0156 0157 0177 0158 0156 0177 0178 0177 0178 0177 0178 0177 0177	* KEYBO * KREAL KREAL KREAL * SERI * SERI * SERI * SERAL SREAL SREAL SREAL SREAL SREAL * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * * * * * SEROT SDROT * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA IN COL S AND SOL IMPLEMENT PUSH	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SOUTPUT H SSERST SDATA A,B SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE AN KETTLEBO OF LEGE STATIO ED CONTAIN ATION FEATU H	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT WAIT FOR INPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS SAVE EVERYBODY
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C031 C032 C036 C036 C036 C036 C036 C036 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C037 C0 C030 C036 C036 C036 C036 C036 C037 C0 C037 C0 C030 C0 F9 C040 C048 C0	0122 0123 0122 0123 0124 0125 0126 0126 0127 0128 0130 0131 0132 0133 0134 0132 0133 0134 0135 0136 0141 0142 0143 0144 0145 0146 0147 0155 0156 0157 0158 0159 0150 0159 0150 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0159 0150 0151 0159 0150 0151 0159 0150 0151 0159 0150 0151 0159 0160 0177 0178 0177 0178 0177 0178 0177 0177	* KEYBO * KREA1 KREA1 KREA1 SSTAT * * * SSTAT * * * SSTAT * * * * * SREA1	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RAL JNC MOV OUT RET IONS. O TER B AND E 'CONSOL MENTATION I COL S AND SOL IMPLEMENT PUSH PUSH PUSH	RETURN THE H KSTAT KREAD KASTAT KREAD KDATA STATUS CHEC SERST SDR · INPUT H SSTAT SREAD SDATA · OUTPUT H SERST · SDROT A,B SDATA · VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE AN KETTLEBO OF LEGE STATIO ED CONTAIN ATION FEATU H D B	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT MAIT FOR IMPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RES. SAVE EVERYBODY ENTRY FROM DEVICE SELECT
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C030 C031 C036 C036 C036 C036 C036 C036 C037 C040 C048	0122 0122 0123 0124 0125 0126 0126 0127 0128 0130 0131 0132 0130 0131 0132 0133 0134 0132 0133 0136 0137 0138 0139 0140 0130 0131 0142 0143 0144 0145 0149 0150 0155 0156 0157 0158 0155 0156 0157 0158 0159 0150 0155 0156 0157 0158 0159 0150 0151 0152 0155 0156 0157 0158 0159 0150 0151 0157 0158 0159 0150 0151 0157 0158 0159 0150 0151 0157 0158 0159 0150 0151 0157 0158 0156 0157 0158 0159 0150 0151 0157 0158 0156 0157 0158 0156 0157 0158 0159 0150 0150 0157 0158 0159 0150 0157 0158 0159 0150 0157 0158 0159 0150 0161 0177 0178 0177 0177	* KREAL KREAL KREAL KREAL KREAL SSTAT * * SER SSTAT * * SER SSREAL SREAL	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP IN CALL JNC MOV OUT RET IONS. O TER B AND E 'CONSOL MENTATION I COL S AND SOL IMPLEMENT PUSH PUSH PUSH PUSH PUSH	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT SSTAT SSTAT SSTAT STAT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET FORT STATUS PUT HIGH BIT IN CARY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RES. SAVE EVERYBODY ENTRY FROM DEVICE SELECT SAVE IN BSTRIP PARITY BEFORE
C027 C027 C027 C027 C028 C030 C031 C031 C031 C031 C031 C031 C031 C031 C031 C035 C036 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 C04 C04 C04 C040 C04	0122 0122 0123 0124 0125 0126 0126 0126 0127 0128 0129 0130 0132 0132 0132 0131 0132 0133 0134 0132 0133 0134 0132 0133 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0132 0133 0134 0132 0134 0132 0132 0133 0134 0132 0134 0132 0134 0132 0134 0132 0134 0132 0134 0132 0134 0132 0134 0132 0134 0132 0134 0134 0137 0138 0141 0142 0155 0156 0157 0156 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0177 0158 0157 0158 0157 0177 0158 0157 0177 0158 0157 0177 0158 0157 0177 0178 0177 0178 0177 0178 0177 0178 0177 0177	* KEYBO * KREAL KREAL KREAL * SERI * SERI * SERAL SREAL SREAL SREAL SREAL SREAL * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * * * SEROT SDROT * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RAL JNC MOV OUT RET I COL S AND SOL IMPLEMENT PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR INPUT H SSTAT STATUS CHEC SERST SDR OUTPUT H SERST SDATA OUTPUT H SERST SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE AN KETTLEBO COF LEGE STATIO ED CONTAIN ATION FEATU H D B PSW A, B H, TBL	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RES. SAVE EVERYBODY ENTRY FROM DEVICE SELECT SAVE IN BSTRIP PARITY BEFORE
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C032 C036 C036 C036 C036 C036 C036 C037 C040	0122 0122 0123 0126 0126 0126 0126 0127 0128 0129 0130 0131 0132 0131 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0141 0142 0143 0144 0145 0146 0147 0148 0149 0150 0151 0152 0156 0157 0156 0157 0158 0159 0150 0151 0152 0155 0156 0157 0158 0159 0150 0151 0152 0155 0156 0157 0158 0157 0158 0159 0150 0151 0152 0156 0157 0158 0159 0150 0151 0152 0156 0157 0158 0157 0177 0158 0177 0178 0177 0178 0178 0177 01718 0177 0178 0177 0178 0177 0178 0177 0178 0177 0177	* KEYBO * KREAL KREAL KREAL * SERI * SERI * SERI * SERAL SREAL SREAL SREAL SREAL SREAL * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * SEROT SDROT * * * * * * * * * * * * * * *	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RAL JNC MOV OUT RET IONS. O TER B AND I COL S AND SOLI IMPLEMENT PUSH PUS	RETURN THE H KSTAT KSTAT KTATUS CHEC SERST SDR STATUS CHEC SERST SDR INPUT H SSTAT STAT STATUS CHEC STAT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RES. SAVE EVERYBODY ENTRY FROM DEVICE SELECT SAVE IN BSTRIP PARITY BEFORE GO PROCESS
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C029 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C035 C036 C036 C036 C036 C036 C036 C037 C040 C040 C040 C040 C040 C040 C048	0122 0123 0124 0125 0126 0126 0126 0127 0128 0127 0128 0130 0131 0132 0131 0132 0133 0134 0135 0136 0137 0138 0139 0141 0142 0133 0144 0145 0147 0148 0147 0148 0147 0148 0147 0148 0141 0142 0151 0151 0152 0153 0154 0155 0156 0157 0158 0159 0150 0151 0152 0155 0156 0157 0158 0159 0150 0151 0152 0155 0156 0157 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0152 0155 0156 0157 0158 0159 0150 0151 0158 0159 0150 0151 0152 0155 0156 0157 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0155 0156 0157 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0157 0158 0159 0150 0151 0158 0157 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0157 0158 0159 0150 0151 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0158 0157 0177 0178 0179 0170 0171 0177 0178 0179 0178 0179 0170 0177 0178 0179 0170 0179 0170 0177 0178 0179 0179 0179 0170 0177 0178 0179 0179 0179 0179 0179 0179 0179 0179	* KEYBO * KREAL KREAL KREAL SSTAT * SERI * SERI SREAL SREAL SREAL SREAL SREAL SREAT SREAT SREAT SEROT SDROT * * * SER * SER	ARD. ON POP CALL JNZ IN RET AL INPUT IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL S AND SOL IMPUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH CALL CALL MOV	RETURN THE H KSTAT KSTAT KSTATUS CHEC SERST SDR STATUS CHEC SERST SDR INPUT H SSTAT SSTAT STATUS CHEC SERST OUTPUT H SERST A,B SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE ANTON FEATU H D B PSW A,B H,TBL TSRCH VDADD A,M '	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR INPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RS. SAVE EVERYBODY ENTRY FROM DEVICE SELECT SAVE IN BSTRIP PARITY BEFORE GO PROCESS GET SCREEN ADDRESS
C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C035 C9 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C036 C036 C036 C036 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 C036 C036 C036 C036 C036 C036 C036 C037 C0 C037 C0 S7 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0	0122 0123 0122 0123 0124 0125 0126 0126 0127 0128 0130 0131 0132 0130 0131 0132 0133 0134 0135 0136 0137 0138 0141 0142 0143 0144 0145 0146 0147 0148 0155 0156 0156 0157 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0156 0157 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0151 0158 0159 0150 0157 0158 0159 0150 0157 0158 0159 0150 0157 0158 0159 0150 0157 0158 0159 0177 0158 0159 0177 0178 0159 0177 0178 0160 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0150 0177 0178 0150 0177 0178 0150 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0151 0177 0178 0150 0177 0178 0150 0177 0178 0150 0177 0178 0150 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0177	* KEYBO * KREAL KREAL KREAL SSTAT * SERI * SERI SREAL SREAL SREAL SREAL SREAL SREAL SREAL SREAL * SER * SEROT SDROT * SEROT * SOLO * SOLO	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANII RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JZ IN RET IAL DATA POP IN RAL JNC MOV OUT RET RET ROUTINES COL S AND SOL IMPLEMENT PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR STATUS CHEC SERST SDR UNPUT H SSTAT SSEAD SDATA OUTPUT H SSDATA OUTPUT ALLOW FOR ALLOW FOR NENTRY, TH ALL REGIST ' VERSION O OF ROUTINE ALLOW FOR NENTRY, TH ALL REGIST ' VERSION O OF ROUTINE AN KETTLEBO OF LEGE STATIO ED CONTAIN ATION FEATU H D B PSW A, B SCH H, TBL TSRCH VDADD A,M 80H M,A	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT MAIT FOR IMPUT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RES. SAVE EVERYBODY ENTRY FROM DEVICE SELECT SAVE IN BSTRIP PARITY BEFORE GO PROCESS GET SCREEN ADDRESS CURSOR IS BACK ON
C027 C027 C027 C027 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C028 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C031 C035 C9 C036 C036 C036 C036 C036 C036 C037 C0 31 C0 C037 C0 31 C0 C037 C0 31 C0 31 C0 C036 C036 C036 C036 C037 C0 31 C0 31 C0 C037 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 31 C0 36 C0 36 C0 36 C0 36 C0 36 C0 37 C0 31 C0 C0 31 C0 20 37 C0 C0 30 C0 37 C0 C0 30 C0 40 C0 C0 C0 C0 C0 C0 C0 C0 C0 C	0122 0123 0122 0123 0124 0125 0126 0126 0127 0128 0130 0131 0132 0133 0134 0132 0133 0134 0132 0133 0134 0132 0133 0134 0137 0138 0139 0141 0142 0143 0144 0145 0146 0147 0148 0149 0150 0151 0155 0156 0157 0158 0155 0166 0167 0172 0153 0161 0177 0188 0159 0150 0161 0177 0188 0159 0150 0160 0177 0178 0150 0160 0177 0178 0150 0160 0177 0178 0160 0177 0178 0160 0177 0178 0160 0177 0178 0160 0177 0178 0160 0177 0178 0160 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0178 0177 0177	* KEYBO * KREAL KREAL KREAL SSTAT * SERI SSTAT * SERAL SREAL SREAL SREAL SREAL SREAT SREAT * SEROT SDROT * SOLO * SEROT * SOLO * SO	ARD. ON POP CALL JNZ IN RET AL INPUT IN ANI RET IAL DATA POP CALL JZ IN RET IAL DATA POP CALL JNC MOV OUT RET IAL DATA POP CALL JNC MOV OUT RET IAL DATA POP IN RAL JNC MOV OUT RET IAL DATA POP IN RAL JNC MOV OUT RET IAL DATA POP IN RAL JNC MOV OUT RET IAL DATA POP IN RAL JNC MOV OUT RET IAL DATA POP IN RAL JNC MOV OUT RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA POP IN RET IAL DATA POP IN COL S AND SOL IMPLEMENT PUSH PU	RETURN THE H KSTAT KREAD KDATA STATUS CHEC SERST SDR STATUS CHEC SERST SDR NPUT H SSTAT SSEAD SDATA OUTPUT H SEEST SDROT A,B SDATA VIDEO DISPL ALLOW FOR N ENTRY, TH ALL REGIST ' VERSION O OF ROUTINE AN KETTLEBO OF ROUTINE AN KETTLEBO COF ROUTINE AN KETTLEBO DE CONTAIN ATION FEATU H D B SB SB A,B H,TBL TSRCH VDADD A,M B SU SU SU SU SU SU SDROT A,B SDATA SDROT A,B SDROT A,B SDROT A,B SDROT A,B SDROT A,B SDROT A,B SDROT A,B SDROT A,B SDROT A,B SDROT	CHARACTER IS IN REGISTER 'A'. JUMP TABLE ENTRY POINT CHECK STATUS WAIT FOR IMPUT GET DATA GO BACK WITH IT K GET SERIAL STATUS WORD TEST FOR SERIAL DATA READY FLAGS ARE SET RESTORE HL FROM JUMP TABLE ENTRY NORMAL ENTRY POINT GET DATA BYTE WE HAVE IT JUMP TABLE ENTRY POINT GET PORT STATUS PUT HIGH BIT IN CARRY LOOP UNTIL TRANSMITTER BUFFER IS GET THE CHARACTER BACK SEND IT OUT AND WE'RE DONE AY DRIVER ROUTINES STANDARD VIDEO TERMINAL E CHARACTER FOR OUTPUT IS IN ERS ARE UNALTERED ON RETURN. F THIS ROUTINE IS A MINIMUM S ORIGINATED BY: ROUGH N, TEXAS THE ESC SEQUENCES AND OTHER RES. SAVE EVERYBODY ENTRY FROM DEVICE SELECT SAVE IN BSTRIP PARITY BEFORE GO PROCESS GET SCREEN ADDRESS CURSOR IS BACK ON

	C9			0193		RET		EXIT FROM VDMOT
C062 C062				0194 0195	*			
C062	7E B7			0196 0197	TSRCH	MOV	A,M A	GET CHR FROM TABLE
C064	CA	7B	C0	0198		JZ	CHAR	ZERO IS THE LAST
C067	<u>в</u> в 23			0199		INX	н	POINT FORWARD
C069 C060	C2 E5	76	C0	0201 0202		JNZ PUSH	NEXT H	FOUND ONE SAVE ADDRESS
C06E C070	CD E1	3D	C1	0203		CALL	CREM H	REMOVE CURSOR
C071				0205	*			
C071				0200	* THIS	S ROUTINE	DISPATCHES	TO THE ADDRESS POINTED TO
C071 C071				0208 0209	* BY THI * LAST I	E HL REGI: ENTRY ON 7	STER PAIR. THE STACK.	THE RETURN ADDRESS IS THE
C071 C071	7E			0210 0211	* DISPT	MOV	A,M	GET LOW BYTE
C072	23			0212		INX	Н И М	אור דעד עד מע
C074	6F			0213		MOV	L,A	WE HAVE PLACED THEM BOTH
C075	E9			0215 0216	*	PCHL	•	GO TO IT
C076 C076	23			0217 0218	* NEXT	INX	н	GO TO NEXT
C077	23 C3	62	CO	0219		INX	H TSRCH	
C07E	0.5	02		0221	*	01112	1011011	
C07E	78			0222	CHAR	MOV	A,B	GET CHARACTER
C07C C07E	B7 C8			0224 0225		ORA RZ	A	RETURN IF A NULL
C07E	FE C8	7F		0226		CPI RZ	7FH	IS IT A DEL? GO BACK IF SO
C081				0228	*			
C081		0.1	C1	0230	*			
C081	78	21	CI	0231	UCHAR	MOV	A,B	GET CHARACTER FOR OUTPUT
C085 C087	E6 77	7F		0233 0234		ANI MOV	7FH M,A	NO HIGH BITS GO PAST HERE PUT CHR ON SCREEN
C088 C08¤	3A FE	00 3F	C8	0235 0236		LDA CPI	NCHAR 63	GET CHARACTER POSITION END OF LINE?
COSE	DA	AD 01	C0	0237		JC LDA	OK	
C090	FE	01 OF		0238		CPI	15	END OF SCREEN?
C095 C098	C2	AD	CU	∪240 0241	*	JNZ	UK.	
C098 C098				0242 0243	* END (OF SCREEN	ROLL UP (ONE LINE
C098	AF 32	00	C8	0244	SCROLL	XRA STA	A NCHAR	BACK TO FIRST CHAR POSTTION
C090	4F	00	C0	0245	SROL	MOV	C,A	
COAD	AF	28	CI	0247 0248		XRA	A A	CALCULATE LINE TO BE BLANKED
COA1 COA4	CD 3A	E9 02	C0 C8	0249 0250		CALL LDA	CLIN1 BOT	CLEAR IT
COA7	3C F6	0 ਜ		0251		INR	A OFH	
COAA	C3	DD	C0	0252		JMP	ERAS3	
COAL				0254	* INCRI	EMENT LIN	E COUNTER IN	F NECESSARY
C0AE C0AE	3A	00	C8	0256 0257	* OK	LDA	NCHAR	GET CHR POSITION
COB0 COB1	3C 32	0.0	C8	0258		INR	A NCHAR	STORE THE NEW
COB4	FE	40		0260		CPI	64	
COBO COB7	AF			0261		XRA	A	WE'RE PAST THE END REWIND THE COUNT
COBS	32 3A	00	C8 C8	0263		STA LDA	NCHAR LINE	GET THE LINE COUNT
COBE COBE	3C E6	0F		0265 0266		INR ANI	A OFH	MOD 15 INCREMENT
C0C1	32	01	C8	0267	CUR	STA	LINE	STORE THE NEW
COCA	C 9			11768		PFT		
C0C4 C0C5	C9			0268	*	RET		
C0C4 C0C5 C0C5 C0C5	C9			0268 0269 0270 0271	* * ERA: *	RET SE SCREEN		
C0C4 C0C5 C0C5 C0C5 C0C5 C0C5	C9 21 36	00 A0	CC	0268 0269 0270 0271 0272 0273	* ERA: * PERSE	RET SE SCREEN LXI MVI	H,VDMEM M,80H+' '	POINT TO SCREEN THIS IS THE CURSOR
C0C4 C0C5 C0C5 C0C5 C0C5 C0C8 C0C8	C9 21 36 23	00 A0	CC	0268 0269 0270 0271 0272 0273 0274 0275	* ERAS * PERSE * ERAS1	RET SE SCREEN LXI MVI INX	H,VDMEM M,80H+' ' H	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT
C0C4 C0C5 C0C5 C0C5 C0C5 C0C8 C0C8 C0C8 C0C8	C9 21 36 23 7C	00 A0	сс	0268 0269 0270 0271 0272 0273 0274 0275 0276	* ERA: * PERSE * ERAS1	RET SE SCREEN LXI MVI INX MOV GDI	H,VDMEM M,80H+' ' H A,H ODOU	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOD
C0C4 C0C5 C0C5 C0C5 C0C8 C0C8 C0C8 C0C8 C0C8	C9 21 36 7C FE D2	00 A0 D0 D6	сс	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278	* ERAS * PERSE * ERAS1	RET SE SCREEN LXI MVI INX MOV CPI JNC	H,VDMEM M,80H+''' H A,H OD0H ERAS2	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP
C0C4 C0C5 C0C5 C0C5 C0C5 C0C8 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4	C9 21 36 7C FE D2 36 C3	00 A0 D0 D6 20 CA	CC C0 C0	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280	* ERA: * PERSE * ERAS1	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP	H,VDMEM M,80H+''' H A,H 0D0H ERAS2 M,''' ERAS1	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK
C0C4 C0C5 C0C5 C0C5 C0C8 C0C8 C0C8 C0C8 C0C8	C9 21 36 7C FE D2 36 C3 AF	00 A0 D0 20 CA	CC C0 C0	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282	* ERAS PERSE * ERAS1	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA	H,VDMEM M,80H+''' H A,H ODOH ERAS1 A	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK
C0C4 C0C5 C0C5 C0C5 C0C8 C0C8 C0C8 C0C8 C0C8	C9 21 36 7C FE D2 36 C3 AF 32 32	00 A0 D0 D6 20 CA 01 00	CC C0 C0 C8 C8	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282 0283 0283	* ERAS PERSE * ERAS1	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA	H,VDMEM M,80H+''' H A,H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SUP OF SCREEN
C0C4 C0C5 C0C5 C0C5 C0C6 C0C6 C0C6 C0C6 C0C6	21 36 7C FE D2 36 C3 AF 32 32	00 A0 D0 D6 20 CA 01 00	CC C0 C8 C8	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282 0283 0284 0284 0285	* ERAS PERSE * ERAS1	RET SE SCREEN LXI MVI INX MOV CPI JNC JNC JNC JNC XRA STA STA STA	H,VDMEM M,80H+''' H A,H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL DARAMETERS
C0C4 C0C5 C0C5 C0C5 C0C8 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4	21 36 23 70 FE 23 6 36 C3 AF 32 32 32	00 A0 D0 D6 20 CA 01 00 FE 02	CC C0 C0 C8 C8 C8	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0277 0278 0279 0280 0282 0282 0283 0284 0285 0284 0285 0286	* ERAS PERSE * ERAS1	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA OUT STA	H,VDMEM M,80H+''' H A,H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESST SCROLL PARAMETERS BEGINNING OF TEXT OFFSET
C0C4 C0C5 C0C5 C0C5 C0C8 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4	C9 21 36 23 7C FE D2 36 C3 AF 32 32 D3 32 C9	00 A0 D0 D6 20 CA 01 00 FE 02	CC C0 C0 C8 C8 C8	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0277 0278 0281 0282 0281 0282 0283 0284 0285 0286 0287 0288 0289	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3	RET SE SCREEN LXI MVI INX MOV CPI JNC CPI JNC STA STA STA OUT STA RET	H,VDMEM M,80H+''' H A,H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET
C0C4 C0C5 C0C5 C0C5 C0C8 C0C8 C0C8 C0C8 C0C8	C9 21 36 7C FE D2 36 C3 AF 32 32 C9 CD	00 A0 D0 D6 20 CA 01 00 FE 02 21	CC C0 C0 C8 C8 C8 C8 C1	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0277 0278 0280 0281 0282 0283 0284 0285 0286 0287 0288 0289 0290 0291	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3	RET SE SCREEN LXI MVI INX MOV CPI JNC CPI JNC STA STA STA OUT STA RET CALL	H, VDMEM M, 80H+' ' H A, H ODOH ERAS2 M,' ' ERAS1 A LINE NCHAR DSTAT BOT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS
C0C4 C0C5 C0C5 C0C5 C0C8 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4 C0C4	C9 21 36 23 7C 7C 2 5 6 2 36 C3 32 32 2 32 C9 CD 3A FE	000 A0 D0 D6 200 CA 01 00 FE 02 21 00 40	CC C0 C0 C8 C8 C8 C8 C8 C1 C1	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282 0283 0284 0285 0286 0287 0288 0288 0289 0291 0291 0292	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLIN1	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA STA STA RET CALL LDA CPI	H, VDMEM M, 80H+' ' A, H 0D0H ERAS2 M,' ' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63
C0C4 C0C5 C0C5 C0C5 C0C6 C0C6 C0C6 C0C6 C0C6	C9 21 36 23 7C 7C 7C 23 6 C3 32 32 C9 CD 3A FE 00 3A FE 00 3A 5 2 3 2 32 32 32 32 32 32 32 32 32 32 32	00 A0 D0 D6 20 CA 01 00 FE 02 21 00 40	CC C0 C0 C8 C8 C8 C8 C8 C1 C8	0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282 0283 0284 0285 0286 0287 0288 0289 0280 0281 0282 0283 0284 0290 0291 0292 0293 0294 0294	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLIN1	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA STA STA STA RET CALL LDA CPI RNC CPI RNC	H, VDMEM M, 80H+' ' A, H 0D0H ERAS2 M,' ' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE
C0C4 C0C5 C0C5 C0C5 C0C6 C0C6 C0C6 C0C6 C0C6	C9 21 36 23 7C FE D2 36 C3 AF 52 32 32 C9 CD 3A FE 00 3A FE 23 32 23 22 20 20 20 20 20 20 20 20 20 20 20 20	00 A0 D6 20 CA 01 00 FE 02 21 00 40 20	CC C0 C0 C8 C8 C8 C8 C8 C8	02689 02699 0270 02712 0272 0273 0274 0275 0276 0277 0278 0277 0280 0280 0280 0282 0283 0284 0285 0288 0288 0288 0288 0288 0288 0288	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLIN1	RET SE SCREEN LXI MVI INX CPI JNC CPI JNC MVI JMP XRA STA STA STA STA CUT STA RET CALL LDA CPI RNC CPI RNC MVI INX	H, VDMEM M, 80H+' ' A, H 0D0H ERAS2 M,' ' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,' ' H	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT
	C9 21 36 23 7C FE 36 C3 32 32 32 C9 CD 3A FE 23 36 23 32 C9 CD 3A FE 23 36 C3 C2 C2 C3 C2 C2 C3 C2 C3 C2 C3 C2 C3 C2 C2 C3 C2 C2 C3 C2 C3 C2 C3 C2 C3 C2 C2 C3 C2 C2 C3 C2 C3 C2 C3 C2 C3 C2 C2 C3 C2 C2 C3 C2 C2 C2 C3 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2	000 A00 D66 200 CA 011 000 FE2 002 201 200 200 200 200	CC C0 C0 C8 C8 C8 C1 C8 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0	0269 0270 0271 0272 0273 0276 0276 0276 0277 0278 0277 0278 0277 0279 0270 0277 0278 0277 0278 0277 0278 0270 0280 028	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLIN1	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INX INX INX JMP	H, VDMEM M, 80H+' ' A, H 0D0H ERAS2 M,' ' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,' ' H A CLIN1	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	21 36 23 7C 52 36 C3 32 32 C9 CD 36 32 32 C9 CD 36 32 32 C9 CD 36 32 32 C9 CD 36 C3 32 C9 CD 23 C0 C3 C3 C2 C3 C2 C3 C2 C3 C3 C2 C3 C2 C3 C2 C3 C3 C2 C2 C3 C2 C2 C2 C2 C2 C2 C3 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2	000 A0 D0 20 CA 01 000 FE 02 21 000 40 20 E9	CC C0 C0 C8 C8 C8 C8 C8 C1 C8 C1 C2 C0	0268 0269 0270 0271 0272 0273 0274 0275 0274 0276 0276 0277 0280 0281 0283 0284 0283 0284 0285 0285 0286 0287 0280 0285 0286 0287 0280 0290 0290 0290 0290 0290 0290 0290	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLIN1	RET SE SCREEN LXI MVI INX MOV CPI JNC CPI JNC STA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INX INX INX INX STA	H, VDMEM M, 80H+''' A, H OD0H ERAS2 M,'' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 23 7C 5 23 6 C3 32 32 C9 CD 3A FE D0 3A FE D0 3A FE C3 3C C3 C3 22 3C C3 23 C3 22 C9 C0 23 C3 C3 C0 C3 C3 C3 C0 C3 C3 C3 C0 C3 C3 C3 C0 C3 C3 C3 C0 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C2 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3	00 A0 D0 D6 20 CA 01 00 FE 02 21 00 40 20 E9	CC C0 C0 C8 C8 C8 C8 C8 C1 C8 C1 C0	0268 0269 0270 0271 0272 0273 0274 0275 0274 0276 0277 0270 0270 0280 0281 0283 0284 0285 0285 0285 0286 0287 0285 0286 0287 0280 0290 0290 0290 0290 0290 0290 0290	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLIN1 * HOMI	RET SE SCREEN LXI MVI INX MOV CPI JNC CPI JNC STA STA STA STA STA STA CUT STA RET CALL LDA CPI RNC CPI INX INX INX INX INX STA STA	H, VDMEM M, 80H+''' A, H 0D0H ERAS2 M,'' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 7C 7C 72 72 72 72 72 72 72 72 72 72 72 72 72	000 A0 D0 D6 20 CA 01 000 FE 02 21 000 40 20 E9	CC C0 C8 C8 C8 C8 C1 C8 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0	0268 0269 0270 0271 0272 0273 0274 0275 0274 0275 0276 0276 0277 0280 0279 0280 0281 0283 0284 0285 0286 0287 0286 0287 0280 0281 0282 0285 0286 0287 0280 0280 0281 0282 0285 0286 0287 0280 0280 0280 0280 0280 0280 0280	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLINE CLIN1 * HOMI	RET SE SCREEN LXI MVI INX MOV CPI JNC CPI JNC STA STA STA STA STA CALL LDA CPI RNC MVI INX INR JMP E CURSOR XRA STA STA	H, VDMEM M, 80H+''' A, H OD0H ERAS2 ERAS2 LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1 A NCHAR CUIN1	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 7C 7C 23 36 7C C3 32 32 32 C9 CD 3A FFE 23 3C C3 3C C3 AF 32 23 3C C3 3C C3 3C C3	00 A0 D6 20 CA 01 00 FE 02 21 00 40 20 E9 00 C1	CC C0 C0 C8 C8 C8 C1 C8 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0	0268 0269 0270 0271 0272 0273 0274 0275 0274 0276 0277 0280 0281 0283 0284 0285 0285 0285 0286 0287 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0280 0281 0282 0285 0286 0287 0280 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0280 0280 0281 0282 0285 0280 0280 0280 0280 0280 0280	* ERAS PERSE * ERAS1 * ERAS2 * ERAS3 * CLINE CLIN1 * HOMI * HOME	RET SE SCREEN LXI MVI INX MOV CPI JNC CPI JNC STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INX INR JMP E CURSOR XRA STA JMP	H, VDMEM M, 80H+''' A, H 0D0H ERAS2 M,'' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,'''H A CLIN1 A NCHAR CUR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 23 7C 7C 23 32 32 C9 23 32 C9 23 32 C9 23 34 FE 23 32 C9 23 32 C9 23 32 C9 23 32 C9 23 32 C9 23 32 C0 23 32 C0 23 23 23 23 23 23 23 23 23 23 23 23 23	00 A0 D0 20 CA 01 00 FE 02 21 00 40 20 E9 00 C1	CC C0 C8 C8 C8 C1 C8 C0 C0 C0 C8 C0	0268 0269 0270 0271 0272 0273 0274 0275 0274 0276 0276 0277 0280 0270 0280 0281 0283 0284 0283 0285 0285 0286 0287 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0280 0281 0282 0285 0286 0287 0280 0280 0281 0283 0285 0286 0287 0280 0280 0280 0280 0280 0280 0280	* ERAS PERSE *ERAS1 *ERAS1 *ERAS2 *ERAS3 *CLINE CLINI CLIN1 * HOMI PHOME	RET SE SCREEN LXI MVI INX MOV CPI JNC CPI JNC STA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INX INR JMP E CURSOR XRA STA JMP	H, VDMEM M, 80H+''' A, H 0D0H ERAS2 ERAS2 LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,'''H A CLIN1 A NCHAR CUR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 23 7C FE 23 36 C3 32 C3 32 C9 CD 33 7C C9 CD 33 7C C9 CD 33 7C C9 C0 33 7C C9 C0 33 7C C9 C0 32 32 C9 C0 32 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3	000 A00 D00 CA 01000 FE2 000 200 200 200 200 200 200 000 C1 001 00F	CC C0 C8 C8 C1 C0 C1 C0 C0 C8 C0 C8 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0	0268 0269 0270 0271 0272 0273 0274 0275 0274 0276 0276 0277 0280 0279 0280 0281 0283 0284 0283 0285 0285 0286 0287 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0283 0285 0286 0287 0280 0281 0282 0283 0285 0286 0287 0280 0281 0283 0285 0286 0287 0280 0281 0282 0283 0285 0286 0287 0280 0281 0283 0285 0286 0287 0280 0280 0281 0283 0285 0286 0287 0280 0280 0280 0280 0280 0280 0280	* ERAS PERSE *ERAS1 *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME	RET SE SCREEN LXI MVI INX MOV CPI JMC MVI JMP XRA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INX INR JMP E CURSOR XRA STA JMP URSOR DOWN LDA CPI	H, VDMEM M, 80H+''' A, H 0D0H ERAS2 M,'' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1 A NCHAR CUR NONE LINE LINE 15	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 23 7C 52 36 C3 32 C3 32 C9 CD 33 C9 CD 36 C3 32 C9 CD 36 C3 32 C9 C0 32 C9 C0 32 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3	000 A0 D06 200 CA 010 000 40 20 20 20 20 20 20 20 20 00 C1 00F	CC C0 C8 C8 C1 C0 C1 C0 C1 C0 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1	0268 0269 0270 0271 0272 0273 0274 0275 0274 0275 0276 0276 0277 0280 0279 0280 0281 0283 0285 0285 0285 0285 0286 0287 0285 0286 0287 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0281 0283 0285 0286 0287 0280 0281 0282 0285 0286 0287 0280 0280 0281 0283 0285 0286 0287 0280 0280 0281 0283 0285 0290 0290 0290 0290 0290 0290 0290 029	* ERAS PERSE *ERAS1 *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 CLIN1 * HOMI PHOME * MOVE CU PDOWN	RET SE SCREEN LXI MVI INX MOV CPI JMC CPI JMC STA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INX INR JMP E CURSOR XRA STA JMP URSOR DOWN LDA CPI RNC MVI INX INX INX INX INX INX INX INX INX IN	H, VDMEM M, 80H+''' A, H 0D0H ERAS2 ERAS2 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,'''H A CLIN1 A NCHAR CUR NONE LINE LINE LINE LINE 15	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 23 36 7C 23 36 23 36 23 32 23 32 23 32 20 33 4 FE 23 30 20 30 32 32 20 30 32 32 20 30 32 32 20 34 50 20 32 32 20 36 32 32 32 20 36 32 32 20 36 32 32 32 32 32 32 32 32 32 32 32 32 32	000 A0 D0 200 CA 01 000 FE 21 000 40 20 20 20 20 20 20 20 00 C1 00F C1	CC C0 C8 C8 C1 C0 C1 C0 C1 C0 C1 C2 C0 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2	0268 0269 0270 0271 0272 0273 0274 0275 0276 0276 0277 0280 0280 0285 0285 0285 0285 0285 0285	* ERAS PERSE * ERAS1 * ERAS1 * ERAS2 * ERAS3 * CLINE CLINE CLIN1 * HOMI PHOME * MOVE CU PHOWN	RET SE SCREEN LXI MVVI INX MOV CPI JNC CPI STA STA STA STA STA CALL LDA CPI RNC MVI INX INR JMP E CURSOR XRA STA JMP URSOR DOWI LDA CPI INR JMP	H, VDMEM M, 80H+''' A, H ODOH ERAS2 ERAS2 LINE NCHAR DSTAT BOT VDADD NCHAR 64 NCHAR 64 NCHAR CLIN1 A NCHAR CLIN1 A NCHAR CLIN1 A NCHAR CLIN1 A NCHAR CUR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE
	C9 21 36 23 7C 23 36 23 37 22 36 23 32 23 32 23 32 23 32 23 32 23 32 23 32 23 32 23 32 23 32 23 32 23 6 5 7 5 7 5 7 6 7 7 7 7 7 7 7 7 7 7 7 7	000 A0 D06 200 CA 01 000 FE 02 21 000 40 20 E9 00 C1 00F C1	CC C0 C8 C8 C1 C0 C1 C0 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0	0268 0269 0270 0271 0272 0273 0274 0275 0276 0276 0276 0277 0285 0285 0285 0285 0285 0285 0285 0285	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PDOWN	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA STA STA STA RET CALL LDA CPI RNC MVI INR JMP E CURSOR XRA STA STA RET LDA CPI INR JMP E CURSOR LDA CPI INR STA STA STA STA STA STA STA STA STA STA	H, VDMEM M, 80H+''' A, H ODOH ERAS2 ERAS2 LINE NCHAR DSTAT BOT VDADD NCHAR 64 NCHAR 64 NCHAR 64 NCHAR CLIN1 A NCHAR CLIN1 CLIN1 A NCHAR CLIN1 CL	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN?
	C9 21 36 76 23 76 23 23 23 23 23 23 23 23 23 23 23 23 23	000 A0 D06 200 CA 01 000 FE 20 21 000 40 20 E9 00 C1 00F C1 01	CC C0 C8 C8 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 0269 0270 0271 0272 0273 0274 0275 0276 0276 0277 0280 0270 0280 0285 0285 0285 0285 0285 0285 028	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 * CLINE CLIN1 * HOMI PHOME * MOVE CU PHOWN	RET SE SCREEN LXI MVVI JNC CPI JNC CPI STA STA STA STA STA CALL LDA CPI RNC MVI INX INR JMP E CURSOR XRA STA JMP URSOR DOWI LDA E TO MOVE LDA	H, VDMEM M, 80H+''' A, H ODOH ERAS2 ERAS2 LINE NCHAR DSTAT BOT VDADD NCHAR 64 NCHAR 64 NCHAR 64 NCHAR CLIN1 A NCHAR CLIN1 A NCHAR CLIN1 A NCHAR CUR I I S A CUR THE CURSOR LINE	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT
	C9 21 36 23 7C 7E 22 23 7C 7E 22 23 22 22 22 22 22 22 22 32 22 22 22	000 A0 D0 D26 200 FE 02 21 000 40 20 E9 00 C1 00F C1 01 0F C1	CC C0 C8 C8 C1 C0 C1 C0 C8 C0 C8 C0 C8 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 (0269) (0271) (0272) (0271) (0272) (0273) (0274) (0275) (0276) (0277) (0276) (027	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PHOME	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INX JMP E CURSOR XRA STA JMP URSOR DOWI LDA CPI RZ LDA CPI RZ LDA CPI RZ LDA CPI RZ	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1 A CLIN1 A NCHAR CUR IINE 15 A CUR THE CURSOR LINE	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESST SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 7C 7E 22 23 27 7E 22 23 22 22 20 34 7E 23 22 23 23 22 20 34 7E 23 23 23 23 23 23 23 23 23 23 23 23 23	000 A0 D0 D0 200 CA 01 00 FE2 21 000 40 20 20 20 00 C1 00F C1 01 0F C1	CC C0 C8 C8 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C0 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 (0269) (0271) (0272) (0271) (0272) (0273) (0274) (0275) (0276) (0277) (0276) (027	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PHOME	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INR JMP E CURSOR XRA STA JMP URSOR DOWN LDA CPI RZ URSOR DOWN LDA CPI RZ JMP	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 K N, '' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M, ''' H A CLIN1 A CLIN1 A NCHAR CUR IINE 15 A CUR THE CURSOR LINE A A CUP	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 7C 7E 22 23 27 7E 22 23 23 22 20 34 7E 23 23 23 23 23 23 23 23 23 23 23 23 23	000 A0 D0 D20 CA 01 00 FE 20 20 40 20 20 20 C1 00 C1 01 0F C1 01 C1	CC C0 C8 C8 C1 C0 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C0 C0 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 0267 0271 0272 0271 0272 0271 0272 0273 0274 0275 0276 0277 0278 0276 0277 0278 0276 0277 0278 0276 0277 0278 0278 0278 0278 0285 0286 0287 0286 0287 0286 0287 0286 0287 0299 0200 0291 0282 0293 0294 0290 0291 0292 0293 0294 0290 0291 0292 0293 0294 0295 0296 0297 0298 0290 0291 0292 0293 0294 0295 0296 0297 0298 0290 0291 0292 0293 0294 0295 0296 0297 0298 0290 0291 0292 0293 0294 0295 0296 0297 0298 0299 0290 0291 0292 0293 0294 0295 0296 0297 0298 0299 0290 0291 0292 0293 0294 0295 0296 0297 0298 0299 0290 0291 0292 0293 0294 0295 0296 0297 0298 0299 0290 0291 0292 0293 0299 0290 0291 0292 0293 0299 0290 0291 0292 0293 0294 0295 0296 0297 0298 0299 0290 0291 0292 0293 0294 0295 0294 0295 0294 0295 0296 0297 0298 0299 0299 0290 0291 0295 0294 0295 0294 0295 0294 0295 0294 0295 0294 0295 0294 0295 0294 0295 0294 0299 0299 0299 0290 0295 0296 0299 0290 0291 0292 0293 0294 0295 0296 0295 0296 0295 0296 0295 0296 0299 0290 0295 0296 0295 0296 0295 0296 0295 0295 0295 0296 0295 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0295 0295 0296 0295 0295 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0299 0290 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0296 0295 0295 0295 0295 0296 0295 0295 0295 0295 0296 0295 0295 0295 0295 0295 0295 0295 0295	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PHOME	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA STA STA CUT STA RET CALL LDA CPI RNC MVI INR JMP E CURSOR CALL LDA CPI RNC MVI INR JMP E CURSOR CDU LDA CPI RZ LDA CPI RZ CPI RZ CPI RZ CPI LDA CPI RZ CPI CPI CPI CPI CPI CPI CPI CPI CPI CPI	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1 A CLIN1 A CLIN1 A CLIN1 A CLIN1 LINE LINE 15 A CUR THE CURSOR LINE A CUR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESST SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 7C 7E 20 37 7E 20 32 32 20 32 32 20 34 52 20 34 52 20 34 52 20 34 52 20 34 52 20 34 52 20 34 52 20 32 20 20 32 20 20 20 20 20 20 20 20 20 20 20 20 20	000 A0 D06 200 CA 010 602 20 20 20 20 20 20 C1 00F C1 01 0F C1 C1	CC C0 C8 C8 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9	0268 0271 0272 0273 0274 0275 0276 0277 0273 0274 0275 0276 0277 0278 0276 0277 0285 0286 0287 0285 0286 0287 0288 0289 0290 0300 0301 0302 0303 0304 0305 0306 0307 0308 0309 0311 0312 0313 0314 0312 0321 03221 0323	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PDOWN	RET SE SCREEN LXI MVI INX MOV CPI JMC WVI JMP XRA STA STA STA OUT STA STA COUT STA RET CALL LDA CPI RNC MVI INR JMP E CURSOR LDA CPI RZ INR JMP E TO MOVE LDA CPI RZ JMP E TO MOVE LDA CRA STA JMP CURSOR DOWI	H, VDMEM M, 80H+'' H A, H ODOH ERAS2 M,'' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1 A CLIN1 A NCHAR CUR NONE LINE 15 A CUR THE CURSOR LINE A A CUR THE CURSOR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 7C 7E 20 37 7E 20 37 7E 20 37 7E 20 32 20 32 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 20 32 20 20 20 32 20 20 20 20 20 20 20 20 20 20 20 20 20	000 A0 D06 200 CA 010 602 20 20 20 20 C1 00 C1 01 0F C1 01 0F C1 01 00 C1	CC C0 C8 C8 C1 C0 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C0 C8 C8 C8 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9	0268 0271 0272 0273 0274 0275 0276 0277 0273 0274 0275 0276 0277 0278 0276 0277 0285 0286 0287 0285 0286 0287 0288 0289 0200 0301 0302 0303 0304 0305 0306 0307 0308 0309 0310 0314 0312 0322 0323 0324 0313 0314 0312 0322 0323 0324 0325	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PDOWN * ROUTINI PUP	RET SE SCREEN LXI MVI INX MOV CPI JMC MVI JMP XRA STA STA STA STA CUT STA RET CALL LDA CPI RMC MVI INR JMP E CURSOR LDA CPI RZ INR JMP E TO MOVE LDA ORA RZ JMP CURSOR LEI LDA ORA	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M, '' H A CLIN1 A CLIN1 A NCHAR CUR THE CURSOR LINE 15 A CUR THE CURSOR LINE A CUR TT ONE POSIT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 77C 7E 20 23 77C 23 23 23 23 23 23 23 23 23 23	000 A0 D0 200 CA 01 00 FE 21 00 40 20 20 20 20 20 20 00 20 20 00 C1 00 C1 00 C1 01 00 C1 00 C1	CC C0 C8 C8 C1 C0 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 0271 0272 0273 0274 0275 0276 0277 0272 0273 0274 0275 0276 0277 0278 0270 0280 0281 0285 0286 0289 0290 0300 0291 0300 0301 0302 0303 0304 03056 0307 0308 03014 0312 0313 0314 0312 03221 03223 03224 03225 0326 0327	* ERAS PERSE *ERAS1 *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 CLIN1 * HOMI PHOME * MOVE CU PDOWN * ROUTINI PUP	RET SE SCREEN LXI MVI INX MOV CPI JMC MVI JMP XRA STA STA STA STA OUT STA STA CUT STA RET CALL LDA CPI RMC MVI INR JMP E CURSOR CDI RXA STA JMP URSOR DOWI LDA CPI RZ INR JMP E TO MOVE LDA ORA RZ DCR CURSOR LEI LDA ORA RZ DCP	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,'''H A CLIN1 A CLIN1 A NCHAR CLIN1 A NCHAR CLIN1 CUR THE CURSOR LINE A A CUR THE CURSOR LINE A A CUR TONE POSIT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 7C 7E 20 37 7C 32 20 34 75 20 32 20 34 75 20 32 20 32 20 34 75 20 32 20 32 20 34 75 20 20 34 75 20 20 32 20 20 34 75 20 20 20 20 20 20 20 20 20 20 20 20 20	000 A0 200 CA 010 602 200 200 200 200 200 200 200 200 20	CC C0 C8 C8 C1 C0 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C8 C8 C8 C8 C9 C8 C9 C8 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9	0268 0271 0272 0273 0274 0275 0276 0277 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282 0283 0284 0285 0286 0287 0288 0289 0290 0300 03012 0302 0303 0304 0305 0306 0307 0308 0314 0312 0322 0323 0324 0313 0314 0312 0322 0324 0325 0326 0327 0328	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PDOWN * ROUTINI PUP	RET SE SCREEN LXI MVI INX MOV CPI JMC MVI JMP XRA STA STA STA STA OUT STA STA CUT STA RET CALL LDA CPI RNC MVI INR JMP E CURSOR CALL LDA CPI RXA STA JMP CURSOR DOWI LDA CPI RZ INR JMP CURSOR DOWI LDA CRA RZ JMP CURSOR LEI LDA ORA RZ DCR JMP	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1 A CLIN1 A CLIN1 A CLIN1 A CLIN1 A CLIN1 CUR THE CURSOR LINE A A CUR THE CURSOR LINE A A CUR TONE POSIT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 7C 7E 25 20 37 7C 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 20 34 7C 20 32 20 20 34 7C 20 32 20 32 20 32 20 32 20 20 20 32 20 20 20 32 20 20 32 20 20 20 20 20 20 20 20 20 20 20 20 20	000 A0 D0 200 CA 01 00 E9 00 C1 00 C1 00 C1 01 C1 00 C1 00 00 00	CC C0 C8 C8 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 0271 0272 0273 0274 0275 0276 0277 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0285 0286 0287 0288 0289 0200 0301 0302 0301 0302 0303 0304 0305 0306 0307 0308 0309 0310 0314 0312 0324 0321 0324 0325 0326 0327 0328 0329 0320 0313 0314 0325	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLIN1 * HOMI PHOME * MOVE CU PDOWN * ROUTINI PUP * MOVE CU PLEFT PCUR	RET SE SCREEN LXI MVI INX MOV CPI JMC MVI JMP XRA STA STA STA OUT STA STA CUT STA RET CALL LDA CPI RNC MVI INR JMP E CURSOR CALL LDA CPI RNC MVI INR JMP E CURSOR COM CPI RZ INR JMP COM COM CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CALL LDA CPI CPI CALL LDA CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CALL LDA CPI CPI CPI CALL LDA CPI CPI CPI CPI CPI CPI CPI CPI CPI CPI	H, VDMEM M, 80H+'' H A, H ODOH ERAS2 M,'' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,'' H A CLIN1 A CLIN1 A CLIN1 A CLIN1 A CLIN1 CUR THE CURSOR LINE A A CUR TT ONE POSIT NCHAR A A NCHAR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP
	C9 21 36 23 7C 7E 25 20 37 7C 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 20 34 8 52 20 32 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 32 20 20 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 20 32 32 20 32 32 32 20 32 32 32 32 32 32 32 32 32 32 32 32 32	000 A0 D0 200 CA 01 00 FE 02 20 20 20 20 20 20 20 20 20 20 00 E9 00 C1 00 C1 01 00 C1 00 C1 00 00 00 C1 00 00 C1 00 00 00 00 20 00 00 00 00 00 00 00 00	CC C0 C8 C8 C1 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 0271 0272 0273 0274 0275 0276 0277 0276 0277 0276 0277 0276 0277 0278 0279 0280 0281 0285 0286 0287 0280 0290 0291 0292 0230 0294 0301 0302 0303 0304 0302 0303 0304 0305 0313 0314 0312 0322 0323 0324 0313 0314 0312 0324 0325 0326 0327 0328 0329	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLINI CLINI * HOME * MOVE CU PHOME * NOVE CU PLEFT PCUR * CU	RET SE SCREEN LXI MVI INX MOV CPI JMC MVI JMP XRA STA STA STA OUT STA STA CUT STA CALL LDA CPI RNC MVI INR JMP E CURSOR E CURSOR LDA CPI RZ INR JMP E TO MOVE LDA CRA RZ DCR CRA CRA STA STA STA STA STA STA STA STA CPI INX STA CPI INX STA CPI CALL LDA CPI CPI CALL LDA CPI CPI CPI STA STA STA STA STA STA STA STA STA STA	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLIN1 A CLIN1 A CLIN1 A CLIN1 A NCHAR CUR THE CURSOR LINE A A CUR THE CURSOR LINE A A CUR TT ONE POSIT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP ION
	C9 21 36 23 7C 7C 32 32 C9 23 32 C9 23 34 7 23 2 C9 23 34 7 C9 23 32 C9 23 34 7 C0 3 C0 3 C0 3 C0 3 C0 3 C0 3 C0 2 C0 3 2 C0 3 2 C0 3 2 C0 3 2 C0 3 2 C0 3 C0 2 C0 3 2 C0 3 C0 2 C0 3 C0 2 C0 3 C0 2 C0 3 C0 2 C0 2	000 A0 D06 200 CA 01 000 E9 002 C1 01 C1 01 C1 00 C1 00 00 00 00 00 00 00 00 00 00 00	CC C0 C8 C8 C1 C0 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C0 C8 C8 C0 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8	0268 0271 0272 0273 0274 0275 0276 0277 0278 0279 0270 0271 0272 0273 0274 0275 0276 0277 0288 0280 0281 0285 0286 0287 0280 0281 0282 0283 0280 0290 0200 0291 0200 0292 0230 0301 0302 0301 0311 0312 0313 0314 0315 0322 0322 0322 0322 0322 0322 0322	* ERAS PERSE ERAS1 *ERAS1 *ERAS3 *CLINE CLINI * HOME * HOWE PHOME * MOVE CU PLOWN * NOVE CU PLEFT PCUR * CUI	RET SE SCREEN LXI MVI INX MOV CCPI JNC MVI JMP XRA STA STA OUT STA STA OUT STA CALL LDA CPI RNC MVI INR JMP E CURSOR CPI RXA STA JMP URSOR DOWI LDA CPI RZ INR JMP E TO MOVE LDA CR RZ DCR STA RZ DCR STA RZ DCR RZ CR STA RZ DCR RZ CR STA RZ CR STA STA STA STA STA STA STA STA STA STA	H, VDMEM M, 80H+''' H A, H ODOH ERAS2 M,''' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M,''' H A CLINI A CLINI A CLINI A CLINI A CLINI A CLINE S CUR THE CURSOR LINE A A CUR THE CURSOR LINE A A CUR TT ONE POSIT NCHAR A CUR T ONE POSIT	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP PUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP TION DON'T GO MORE BACK THAN BACK
	C9 21 36 23 7C 7C 7C 23 2 2 2 32 2 2 32 2 2 32 2 2 32 2 3	000 A0 D06 200 CA 000 FE2 21 000 20 20 20 20 20 20 20 00 C1 00F C1 01 00F C1 01 00F C1 00 00 01 00F 02 3F	CC C0 C3 C8 C3 C3 C0 C3 C0 C3 C0 C3 C0 C3 C0 C3 C3 C0 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3	0269 0270 0271 0272 0273 0274 0276 0277 0278 0277 0278 0270 0270 0270 0270	* ERAS PERSE *ERAS1 *ERAS2 *ERAS3 *CLINE CLINI CLINI * HOMI PHOME * MOVE CU PHOME * ROUTINI PUOP * MOVE CU PLEFT PCUR * CUI	RET SE SCREEN LXI MVI INX MOV CPI JNC MVI JMP XRA STA STA OUT STA STA CALL LDA CPI RXC MVI INX INR JMP E CURSOR LDA CPI RZ INR JMP URSOR DOWI LDA CPI RZ INR JMP CURSOR DOWI LDA CRA RZ DCR STA RET INR STA STA STA STA STA STA CPI RZ INR RZ INR STA STA STA STA STA STA STA STA STA STA	H, VDMEM M, 80H+'' H A, H ODOH ERAS2 M, '' ERAS1 A LINE NCHAR DSTAT BOT VDADD NCHAR 64 M, '' H A CLINI A CLINI A CLINI A CLINI A CLINI A CLINE LINE CUR THE CURSOR LINE A CUR THE CURSOR LINE A A CUR THE CURSOR A CUR THE CURSOR A CUR THE CURSOR A CUR THE CURSOR A CUR THE CURSOR A CUR CUR CUR CUR CUR CUR CUR CUR CUR CUR	POINT TO SCREEN THIS IS THE CURSOR BUMP THE COUNT GET HIGH ORDER THE TOP FUT IN A BLANK ZERO LINE LEFT SIDE OF SCREEN RESET SCROLL PARAMETERS BEGINNING OF TEXT OFFSET GET CURRENT SCREEN ADDRESS CURRENT CURSOR POSITION NO MORE THAN 63 ALL DONE ALL SPACED OUT LOOP TO END OF LINE HOW FAR IS DOWN? UP ONE LINE GET LINE COUNT DON'T GO MORE UP THAN UP TION DON'T GO MORE BACK THAN BACK

C11E	C3	13	C1	0337		JMP	PCUR	
C121				0338	* POITT	INF TO C	AT CITLATE SC	
C121				0340	*	INE IO C.	ALCOLATE SC	KEEN ADDRESS
C121				0341	* ENTR	Y AT:	RETURNS:	
C121				0343	*	VDADD	CURRENT SC	REEN ADDRESS
C121				0344	*	VDAD2	ADDRESS OF	CURRENT LINE, CHAR 'C'
C121				0345	*	VDAD	DINE A,	CHARACTER POSITION C
C121	3A 4 ह	00	C8	0347	VDADD	LDA	NCHAR	GET CHARACTER POSITION
C125	3A	01	C8	0349	VDAD2	LDA	LINE	LINE POSITION
C128	6F 3A	02	C 8	0350	VDAD	MOV	L,A BOT	INTO 'L' GET TEXT OFFSET
C12C	85	01	00	0352		ADD	L	ADD IT TO THE LINE POSITION
C12D C12E	0F 0F			0353		RRC		MAKES FOUR
C12F	6F			0355		MOV	L,A	L HAS IT
C130 C132	Е6 57	03		0355		MOV	3 D,A	MOD THREE FOR LATER
C133	3E	CC		0358		MVI	A, <vdmem< td=""><td>LOW SCREEN OFFSET</td></vdmem<>	LOW SCREEN OFFSET
C136	62 67			0360		MOV	H,A	NOW H IS DONE
C137	7D F6	CO		0361		MOV	A,L OCOH	TWIST L'S ARM
C13A	81	00		0363		ADD	C	
C13B	6F (79			0364		MOV	L,A	H & L ARE NOW PERVERTED
C13D				0366	*			
CI3D C13D				0368	* ROU:	FINE TO	REMOVE CURS	OR
C13D	CD	21	C1	0369	CREM	CALL	VDADD	GET CURRENT SCREEN ADDRESS
C140 C141	7E E6	7F		0370		ANI	A,M 7FH	STRIP OFF THE CURSOR
C143	77			0372		MOV	M,A	
C144 C145	09			0374	*	REI		
C145				0375	* R01	JTINE TO	BACKSPACE	
C145	CD	0D	C1	0377	PBACK	CALL	PLEFT	
C148 C14P	CD 36	21 20	C1	0378		CALL MVT	VDADD M.''	GET SCREEN ADDRESS PUT A BLANK THERE
C14D	C9	20		0380		RET	,	
C14E C14E				0381 0382	* ROI	JTINE TO	PROCESS A	CARRIAGE RETURN
C14E	C -		<i>a</i> ^	0383	*		01 131	CLEAD EDON CURDENT CURCCE
CI4E LINE	CD	щЗ	00	U384	PCK	САЦЬ	CLINE	CLEAR FROM CURRENT CURSOR TO END OF
C151	AF	1 2	01	0385		XRA	A	REWIND IT
C152 C155	دى	τŞ	CT	0386	*	UMP	FCUR	AND STOKE THE INEW VALUE
C155				0388	* ROUT	INE TO P	ROCESS LINE	FEED
C155 C155	3A	01	C8	0390	PLF	LDA	LINE	GET LINE COUNT
C158	FE D2	0F	C1	0391		CPI	15	ARE WE AT THE BOTTOM?
C15D	3C	01	CI	0393		INR	A	
C15E C161	C3	C1	C0	0394	*	JMP	CUR	ONE MORE LINE UP
C161	AF	0.0	a 0	0396	SC	XRA	A	
C165	00	50	00	0398	*	0111	BROL	
C165 C165				0399	*			
C165				0401	* TH:	IS TABLE	DEFINES TH	E CHARACTERS FOR SPECIAL
C165 C165				0402	* GOES TO	D THE SC	F THE CHARA REEN.	CTER IS NOT IN THE TABLE IT
C165	88			0404	* TRI.	DB	CLEAR	SUBLEN
C166	C5	C0		0406		DW	PERSE	
C168 C169	97 04	C1		0407		DB DW	UP PUP	CURSOR
C16B	9A	G 0		0409		DB	DOWN	
C16E	81	CU		0410		DB	LEFT	
C16F	0D 93	C1		0412		DW DB	PLEFT	
C172	93 17	C1		0413		DW	PRIT	
C174	8E	G 0		0415		DB	HOME	
C177	0D	CU		0410		DB	CR	CARRIAGE RETURN
C178	4E 0 a	C1		0418		DW DB	PCR	LINE FFFD
C17B	55	C1		0420		DW	PLF	
C17D C17E	5F 45	C1		0421 0422		DB DW	BACKS	BACK SPACE
C180	80			0423		DB	MODE	MODE KEY
C181 C183	B4 00	CI		0424 0425		DW DB	COMND 0	END OF TABLE
C184				0426	*			
C184 C184				0427	* 001151	DI DEVIC	E TABLE	
C184	4C	C0		0429	OTAB	DW DW	VDM01	VDM DRIVER
C188	40 AD	C1		0430		DW	ERROT	ERROR HANDLER (FOR CONSOL)
C18A	AD	C1		0432	*	DW	ERROT	ERROR HANDLER
C18C				0434	* INPU	r device	TABLE	
C18C C18C	27	C0		0435	ITAB	DW	KREA1	KEYBOARD INPUT
C18E	36	CO		0437		DW	SREA1	SERIAL INPUT
C190 C192	ad AD	CI CI		0438 0439		DW DW	ERROT	ERROR HANDLER (FOR CONSOL) ERROR HANDLER
C194				0440	*			
C194 C194				0441 0442	*	COM	MAND TABLE	
C194				0443	* * ידעד כי	יית האצות	פרפופדפ ייניי	VALUE COMMANDS FOR CONSOL
C194	_			0445	*	us UB	оскатор 186	THE CONFILMED FOR CONSUL
C194 C196	54 70	45 C2		0446	COMTAB	ASC DW	'TE' TERM	
C198	44	55		0448		ASC	'DU'	
C19A C19C	45	02 4E		0449		ASC	'EN'	
C19E	07 4F	C3		0451		DW	ENTER	
C1A2	35	C3		0453		DW	EXEC	
C1A4 C1A6	54 39	4C C3		0454		ASC DW	'TL' TLOAD	
C1A8	42	41		0456		ASC	'BA'	SPECIAL COMMAND TO EXECUTE 0
CIAA CIAC	00 00	00		0457 0458		DW DB	U 0	END OF TABLE MARK
CIAD	,			0459	*			
CIAD C1AD				0460 0461	*	C	ONSOL PORT	ERROR HANDLER
CIAD	7.17			0462	*	YPA	Δ	
CIAE	мг [.] 32	04	C8	0464	DIG I UNITED	STA	IPORT	DEFAULT TO SCREEN
C1B1 C1B4	32	03	C8	0465 0466	*	STA	OPORT	DEFAULT TO SCREEN
C1B4				0467	*			
CIB4 C1B4				0468 0469	*			
C1B4				0470	*	=	COMMAND M	ODE=
C1B4				0471	*		0	
C1B4 C1B4				0473 0474	* THIS *	ROUTINE	GETS AND P	RUCESSES COMMANDS
C1B4	31	00	CC	0475	COMND	LXI	SP, SYSTP	SET STACK POINTER
C1BA	CD	⊥∠ C3	C1	0476		CALL	GCLIN	GET COMMAND LINE
C1BD C1C0	CD C3	E1 B4	C1 C1	0478 0479		CALL JMP	COPRC	PROCESS THE LINE OVER AND OVER

0480	*							
0481 0482	*							
0483 0484	* THIS * KEYBO	ROUTINE ARD AND P	READS A COM ROCESSES IT	MAND LINE FROM THE SYSTEM TO THE SCREEN.				
0485 0486	* * C/R	TERMINAT	ES THE SEOU	ENCE ERASING ALL CHARS TO THE				
0487	* * * T/E	RIGHT OF THE CURSOR TERMINATES THE SEQUENCE						
0489	* MODE	RESTARTS	THE COMMAN	D LINE.				
0490	GCLIN	CALL	KREAD	READ KEYBOARD				
0492 0493		MON	20H B,A					
0494 0495	CONT	JC CALL	PROCS VDMOT	PROCESS CONTROL CHARACTER				
0496 0497	*	JMP	GCLIN					
0498	* PROC	ESS CONTR	OL KEYS					
0500	PROCS	CPI	CR	PRACE THE DEMATNING I THE DIAD TO DETIDA				
0501		CPI	LF	ERASE THE REMAINING LINE PRIOR TO RETORN				
0503		JMP	GCLIN	NO CONTROL CHARS TO SCREEN				
0505 0506	*							
0507 0508	CRPRC	CALL RET	CLINE	CLEAR REMAINING LINE NOW PROCESS				
0509 0510	*							
0511 0512	* F *	IND AND P	ROCESS C	OMMAND				
0513	COPRC	CALL	CREM	REMOVE THE CURSOR				
0515		CALL	VDAD2	GET SCREEN ADDRESS				
0510		CALL	SCHR	SCAN PAST BLANKS				
0518		XCHG	· · · · · · · · · · · · · · · · · · ·	HL HAS FIRST CHR				
0520	*	LXI	D, COMTAB	POINT TO COMMAND TABLE				
0522 0523	* THIS * BY 'D	ROUTINE E', FOR A	DOUBLE CHA	ROUGH A TABLE, POINTED TO RACTER MATCH OF THE 'HL'				
0524 0525	* MEMOR * BY PL	Y CONTENT ACING A Q	. IF NO MA UESTION MAR	TCH IS FOUND THE SCAN ENDS K WITHIN THE SEARCH STRING.				
0526 0527	* FDCOM	LDAX	D					
0528		ORA	A FPP2	TEST FOR TABLE END				
0530		PUSH	H	SAVE START OF SCAN ADDRESS				
0532		INX	D	TEST FIRST CIR				
0533	*	JNZ	NCOM					
0535		lnx Ldax	н D					
0537 0538		CMP JNZ	M NCOM	NOW SECOND CHARACTER GOODNESS				
0539 0540	*	POP	в	CLEAR THE STACK				
0541 0542		XCHG INX	н	DE HAS SCAN ADDRESS HL HAS COMMAND ADDRESS				
0543	*	JMP	DISPT	DISPATCH TO IT				
0545	* NCOM	TNY	D					
0547	NCOM	INX	D	GO TO NEXT ENTRY				
0540		POP	H	GET BACK ORIGINAL ADDRESS				
0550	*	JMP	FDCOM	CONTINUE SEARCH				
0551								
0551 0552 0553	* * 0	UTPUT A C	RLF FOLLOWE	D BY A PROMPT				
0551 0552 0553 0554 0555	* * 0 * (WIT *	UTPUT A C H CONSOL	RLF FOLLOWE ALL OPERATI	D BY A PROMPT ONS ARE ON THE SCREEN)				
0551 0552 0553 0554 0555 0556 0557	* 0 * (WIT * PROMPT	UTPUT A C H CONSOL CALL MVI	RLF FOLLOWE ALL OPERATI CRLF B,'>'	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT				
0551 0552 0553 0554 0555 0556 0557 0558 0559	* 0 * (WIT * PROMPT	UTPUT A C H CONSOL CALL MVI JMP	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN				
0551 0552 0553 0554 0555 0556 0557 0558 0559 0560 0560 0561	* 0 * (WIT * PROMPT * CRLF	UTPUT A C H CONSOL CALL MVI JMP MVI CALL	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED				
0551 0552 0553 0554 0555 0555 0555 0557 0558 0559 0560 0561 0562 0563	* 0 * (WIT * PROMPT * CRLF	UTPUT A C H CONSOL MVI JMP MVI CALL MVI JMP	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN DUTT IT OUT AND RETURN				
0551 0553 0553 0554 0555 0556 0559 0559 0560 0561 0562 0563 0563	* 0 * (WIT * PROMPT * CRLF	UTPUT A C H CONSOL MVI JMP MVI CALL MVI CALL MVI JMP	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN				
0551 0553 0554 0555 0556 0556 0556 0558 0560 0561 0562 0564 0565 0564	* * 0 * (WIT * PROMPT * CRLF * * * \$CAN	UTPUT A C H CONSOL MVI JMP MVI CALL MVI JMP OVER UP T	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK				
0551 0553 0554 0555 0556 0557 0558 0559 0560 0562 0563 0564 0565 0566 0565	* O * (WIT PROMPT * CRLF * * * SCAN * SELK	UTPUT A C H CONSOL MVI JMP MVI CALL MVI JMP OVER UP T MVI	RLF FOLLOWE ALL OPERATI CRLF 5,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING				
0551 0553 0554 0555 0556 0557 0558 0559 0561 0562 0563 0564 0565 0566 0566 0566 0566 0566 0566	* O * (WIT PROMPT * CRLF * * * SCAN * SBLK SBLK1	UTPUT A C H CONSOL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI	RLF FOLLOWE ALL OPERATI CRLF ,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING				
0551 0553 0553 0555 0555 0555 0555 0555	* O * (WIT * PROMPT * CRLF * * * SCAN * SBLK SBLK1	UTPUT A C H CONSOL MVI JMP MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT				
0551 0553 0553 0554 0555 0556 0557 0558 0560 0561 0562 0563 0566 0566 0566 0566 0566 0566 0566	* O * (WIT * PROMPT * CRLF * * SBLK SBLK1	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JNZ	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE				
05512 0553 05553 05555 05556 05556 05556 05561 05663 05661 05663 05664 05663 05664 05666 05666 05666 05670 05702 05772 05774 05772 05774	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK1	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JNZ RET	RLF FOLLOWE ALL OPERATI CRLF b,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET				
03512 0553 05553 05556 05557 0558 05560 05661 05663 05664 05665 05666 05668 05666 05670 05673 0574 05774 05774 05774	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK1	UTPUT A C H CONSOL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ INX RET	RLF FOLLOWE ALL OPERATI CRLF b,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SCHR D C SBLK1	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR				
03512 0553 05553 05556 05557 0558 05560 0561 0562 05662 05663 05662 05663 05665 05666 05673 0573 0574 05773 05774 05778 05776 05778 05776	* O * (WIT * PROMPT * CRLF * SCAN * SBLK SBLK1 * * SCAN * A NON *	UTPUT A C H CONSOL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JNZ RET PAST UP T BLANK CHA	RLF FOLLOWE ALL OPERATI CRLF CRLF VDMOT B,CR VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1 0 10 BLANK RACTER.	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR				
03552 05553 05554 05556 05557 05556 05557 05559 05560 05662 05662 05662 05662 05665 05665 05668 05667 05762 05745 05774 05774 05776 05776 05778 05778 05780 05788 05780 05788 05778 05788 05778 05778 05778 05788 05788 05778 05778 05788 05778 05788 05778 05788 05778 05788 05778 05778 05788 05788 05788 05778 05778 05778 05778 05788	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK1 * * SCAN * A NON * SCHR \$ CUP 1	UTPUT A C CALL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ JNZ RET PAST UP T BLANK CHA MVI LDAY	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,CR VDMOT B,CR VDMOT 0 12 CHARAC C,12 D C SBLANK SCHR D C SBLK1 0 10 BLANK RACTER. C,10	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHAPACTER				
03552 05553 05553 05556 05557 05556 05557 05558 05560 05662 05662 05662 05663 05665 05668 05669 05772 05774 05774 05774 05776 05776 05778 05776 05778 05778 05782 05774 05778 057820	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK1 * * SCAN * A NON * SCHR SCHR1	UTPUT A C CALL MVI JMP MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI DCR JNZ RET PAST UP T BLANK CHA MVI LDAX CPI DDY DDY DDY DDY DDY DDY DDY DD	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1 0 10 BLANK RACTER. C,10 D SPACE	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER				
03552 05554 05554 05556 05556 05556 05567 05569 05661 05662 05662 05664 05665 05669 05772 05774 05774 05776 05776 05778 05776 05778 05778 05781 05782 05582 05881 05884 0	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK1 * * * SCAN * * A NON * * SCHR SCHR1	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RNZ INX NX	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1 0 10 BLANK RACTER. C,10 D SPACE 0	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS				
0552 0553 0553 0555 0555 0555 0555 0555	* O * (WIT * PROMPT * * * * * SBLK SBLK1 * * * * * * * * * * * * * * * * * * *	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI JMP OVER UP T LDAX CPI JZ INX DCR JJZ INX DCR JJZ RET PAST UP T BLANK CHA MVI LDAX CPI RDAX RDAX CPI RDAX CPI RDAX RDAX RDAX RDAX RDAX RDAX RDAX RDAX	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1 0 10 BLANK RACTER. C,10 D SPACE D C	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR				
05512 0553 05553 05556 05556 05556 05559 05560 05652 05661 05652 05662 05663 05665 05667 05680 05677 0572 0574 05775 05773 05774 05775 05779 05770 05772 05778 05779 05780 05820 05821 05820 05720 057800 05820 05820 05820 05820 05820 05820 05820 05820 05820 057700 057800 05820 05820 05820 05820 05820 05820 05820 05820 05820 05820 05820 05820 05820 05820 05820 058200 05820 058200 05820 058200 058200 05820000000000	* O * (WIT PROMPT * CRLF * * SCAN * * SCAN * * SCAN * * SCAN * * SELK SBLK1	UTPUT A C H CONSOL MVI CALL MVI AVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ INX DCR JZ INX RET PAST UP T BLANK CHA MVI LDAX CPI LDAX CPI JNZ RET PAST UP T LDAX CPI LDAX CPI JNZ RET	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1 0 10 BLANK RACTER. C,10 D SPACE SCHR1	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR				
03512 0553 05553 05553 05555 05555 05555 05555 05555 05555 05555 05555 05555 05562 05662 05662 05662 05662 05662 05662 05662 05662 05662 05662 05662 05662 0577 05782 0577 05782 05776 05778 05776 05778 05776 05778 05776 05778 05776 05778 05776 05778 05778 05776 05778 05777 05780 05820 05820 05820 05820 05820 05820 05820 05977 05780 05777 05780 05777 05780 05777 05780 05777 05780 05880 05770 05778 05778 05778 05780 05778 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05770 05780 05780 05780 05780 05780 05880 05880 05770 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05780 05890 05890 05990 05890 05990 05890 05990	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK1 * * SCAN * * * SCAN * * * * * * * * * * * * *	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RET PAST UP T BLANK CHA MVI LDAX CPI JNZ RET RNZ INX DCR JMP RNZ TS THE FO	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SBLK1 0 10 BLANK RACTER. C,10 D SPACE SCHR1 CANS OVER C LLOWING ADD	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO				
03512 0553 05553 05553 05555 05555 05555 05555 05555 05555 05555 05555 05555 05555 05555 05555 05555 05565 05662 05662 05662 05665 05662 05665 05677 0573 05742 05775 05775 05775 05775 05775 05778 05780 05772 05778 05780 05823 05884 05882 05884 05885 05884 05885 05884 05885 05884 05885 05577 05575 05577 05775 05775 05775 05777 05785 05777 05785 05775 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05785 05777 05785 05777 05785 05785 05777 05785 05777 05785 05777 05785 05777 05785 05777 05785 05785 05785 05777 05785 05585 05585 05587 05585 05587 05585 05587 05585 05587 05597 05	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK1 * * * * * * * * * * * * *	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ INX DCR PAST UP T BLANK CHA MVI LDAX CPI RET PAST UP T BLANK CHA MVI LDAX CPI RNZ JNZ RET PAST UP T BLANK CHA MVI CPI RNZ SCPI RNZ JNZ RET DCR RZ JMP ROUTINE S TS THE FO ROR HANDL	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT C,12 CHARAC C,12 D BLANK SCHR D C SSBLK1 C,10 D SPACE SCHR1 CANS OVER C CLOWING ADD ER.	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO				
03522 0553 05553 05553 05556 05557 05559 05550 05559 05550 05559 05560 05662 05662 05663 05665 05663 05665 05667 05669 05772 05772 05774 05775 05776 05779 05801 05780 05777 05780 05778 05779 05881 05882 05884 05886 05884 05886 05884 05886 05884 05886 05884 05886 05887 05886 05886 05886 05887 05886 05887 05886 05577 05576 05776 05776 05776 05776 05776 05776 05776 05776 05776 05789 05880 05886 05777 05786 05776 05786 05776 05786 05776 05786 05776 05786 05776 05786 05786 05776 05786 05786 05776 05786 05786 05776 05786 05786 05786 05786 05786 05776 05786 05786 05786 05786 05786 05787 05786 05786 05786 05787 05786 05786 05786 05786 05787 05786 05586 05586 05586 05586 055876 055777 05576 055776 055776 055776 055776 055776 05586 055876 05586 055876 055976	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK SBLK SBLK * * * * * * * * * * * * *	UTPUT A C H CONSOL CALL MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR BLANK CHA MVI LDAX CPI BLANK CHA MVI LDAX CPI BLANK CHA MVI LDAX CPI STS THE FO ROUTINE S TS THE FO ROUTINE S TS THE FO ROUTINE S	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SSBLK1 C,10 D SPACE SCHR1 CANS OVER C LLOWING ADD ER. SBLK FRR1	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO				
03522 0553 05553 05553 05556 05557 05559 05560 05652 05661 05662 05662 05665 05667 05665 05667 05665 05677 05772 05774 05776 05772 05774 05776 05778 05778 05778 05778 05778 05778 05778 05778 05778 05788 05884 05884 05884 05886 05884 05888 05884 05886 05884 05886 05884 05886 05884 05886 05884 05886 05884 05886 05884 05886 05876 05772 05778 05778 05786 05886 05886 05886 05886 05886 05886 05886 05886 05886 05877 05578 05588 05588 05886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055886 055896 0559910 05593 055990 055930 0559910 055930 055930 055930000000000000000000	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK SBLK SBLK * * * CONVER * * THIS * CONVER * * THE ER * * * THE S	UTPUT A C H CONSOL MUI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ INX DCR RET PAST UP T BLANK CHA MVI LDAX CPI RET NZ INZ RET RET RET RET RET RET RET RET RET RET	RLF FOLLOWE ALL OPERATI CRLF b,'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SSBLK1 C,10 D SPACE SCHR1 CANS OVER C LLOWING ADD ER. SBLK ERR1	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO				
03552 0553 05553 05556 05557 05556 05557 05559 05560 05662 05662 05665 05667 05665 05667 05665 05667 05665 05667 05772 05774 05776 05774 05776 05777 05774 05776 05778 05777 05778 05778 05778 05778 05882 058845 058845 058886 058884 058886 058884 05886 058886 058884 05886 058887 05586 05591 05593 05595 05595 05595 05595 05595 05595 05595 05595 05577 05778 05778 05778 05778 05788 05588 05585 05585 05577 05577 05578 05578 05578 05577 05578 05578 05577 05578 05588 05589 0559	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK1 * * * * * * * * * * * * *	UTPUT A C H CONSOL CALL MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR DAX CPI BLANK CHA MVI LDAX CPI BLANK CHA MVI LDAX CPI RXZ DCR RZ JMP ROUTINE S TS THE FO ROR HANDL CALL JZ ROUTINE C CALL JZ ROUTINE C CALL CALL CALC CALL CALC CPI CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC CPI CALC	RLF FOLLOWE ALL OPERATI CRLF b,'>' VDMOT B,LF VDMOT C,12 CHARAC C,12 D BLANK SCHR D C SCHR D C SBLK1 C,10 D SPACE D C SPACE SCHR1 CANS OVER C LLOWING ADD ER. SBLK ERR1 ONVERTS ASC CONVERSION.	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO				
$\begin{array}{c} 0.3522\\ 0.553\\ 0.553\\ 0.553\\ 0.553\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.556\\ 0.557\\ 0.572\\ 0.577\\ 0.577\\ 0.577\\ 0.578\\ 0.578\\ 0.582\\ 0.588\\ 0.599\\ 0.599\\ 0.599\\ 0.599\\ 0.599\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0$	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK1 * * * SCAN * SBLK1 * * * * * * * * * * * * *	UTPUT A C H CONSOL MUI JMP CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ INX DCR JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RNZ INX DCR RZ JMP ROUTINE S TS THE FO ROR HANDL CALL JZ ROUTINE C DARD HEX IS ENCOUN	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,CR VDMOT B,CR VDMOT B,CR VDMOT C. C.12 CARACC C.12 D BLANK SCHR D C SCHR C SELK C. SPACE D C SPACE SCHRI CANS OVER C LLOWING ADD ER. SBLK ERRI ONVERTS ASC CONVERTS	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO				
$\begin{array}{c} 0.352\\ 0.3553\\ 0.553\\ 0.553\\ 0.555\\ 0.556\\ 0.557\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.578\\ 0.5881\\ 0.5883\\ 0.5885\\ 0.5883\\ 0.5885\\ 0.5884\\ 0.5885\\ 0.5884\\ 0.5885\\ 0.5885\\ 0.5885\\ 0.5991\\ 0.5934\\ 0.5991\\ 0.5934\\ 0.5991\\ 0.5934\\ 0.5995\\ 0.5994\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5995\\ 0.5989\\ 0.5996\\ 0.5989\\ 0.5996\\ 0.5989$	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK1 * * * * * * * * * * * * *	UTPUT A C H CONSOL MUI JMP CALL MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ DCR JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RNZ JNZ RET RUTINE S TS THE FO ROUTINE S TS THE FO ROR HANDL CALL JZ ROUTINE C DARD HEX CALL JZ	RLF FOLLOWE ALL OPERATI CRLF S,'>' VDMOT B, LF VDMOT B, CR VDMOT B, CR VDMOT C, 12 C, 12 D BLANK SCHR D C SBLK1 C SBLK1 C, 10 D C SPACE D C SPACE SPACE SCRR1 CANS OVER C LLOWING ADD ER. SBLK ERR1 ONVERTS ASC CONVERTS ASC CONVENTS ASC CONVERTS ASC CONVENTS ASC	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L				
$\begin{array}{c} 0.352\\ 0.553\\ 0.553\\ 0.553\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.555\\ 0.556\\ 0.556\\ 0.557\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.578\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.599\\ 0.593\\ 0.599\\ 0.593\\ 0.598\\ 0.599\\ 0.598\\ 0.598\\ 0.599\\ 0.598\\ 0.$	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK1 * * * * * * * * * * * * *	UTPUT A C H CONSOL MUI JMP CALL MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ DCR JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RNZ INX DCR RZ JMP ROUTINE S TS THE FO ROR HANDL CALL JZ ROUTINE C CALL JZ ROUTINE C CALL JZ ROUTINE C CALL JZ ROUTINE C CALL JZ	RLF FOLLOWE ALL OPERATI CRLF s,'>' VDMOT B,CR VDMOT B,CR VDMOT B,CR VDMOT C C,12 D ELANK SCHR D C SBLK1 C SBLK1 C C,10 D C SPACE D C C SPACE SPACE C SPACE SCHR1 CANS OVER C LLOWING ADD ER. SELK ERR1 ONVERTS ASC CCONVERSION. SELK ERR1 ONVERTS ASC CONVERSION. SELK SCREN WI H,0 D 20H	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE?				
$\begin{array}{c} 0.352\\ 0.553\\ 0.553\\ 0.553\\ 0.555\\ 0.557\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.578\\ 0.588\\ 0.590\\ 0.593\\ 0.595\\ 0.598\\ 0.599\\ 0.593\\ 0.599\\ 0.593\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.599\\ 0.598\\ 0.$	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK1 * * SBLK1 * * * SCAN * SBLK1 * * * * * * * * * * * * *	UTPUT A C H CONSOL MUI JMP CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JJZ INX DCR JNZ RET BLANK CHA MVI LDAX CPI RNZ INX DCR RZ JMP ROUTINE S TS THE FO ROR HANDL JZ ROUTINE S TS THE FO ROR HANDL JZ ROUTINE C DARD HEX CALL JZ ROUTINE C DARD HEX CPI ROUTINE C DARD HEX CPI RE ON TH LDAX CPI RE ON TH LDAX CPI RE ON TH LDAX CPI RE ON TH LDAX CPI RE ON TH LDAX CPI RE ON TH LDAX CPI RE ON TH	RLF FOLLOWE ALL OPERATI CRLF S,'>' VDMOT B, LF VDMOT B, CR VDMOT B, CR VDMOT C, 12 D ELANK SCHR D C SBLK1 C SBLK1 C SPACE D C SPACE SPACE SPACE SPACE SCHR1 CANS OVER C LLOWING ADD ER. SELK ERR1 ONVERTS ASC CONVERSION. TERED. PSC SCASS SCASS CONVERSION. SELK SELK SCONVERTS ASC CONVERSION. SELK SCONVERTS ASC CONVERSION. SELK SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERTS ASC CONVERSION. SCONVERTS ASC CONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERSION. SCONVERTS ASC CONVERTS ASC CONV	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO				
$\begin{array}{c} 0.352\\ 0.553\\ 0.553\\ 0.555\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.566\\ 0.566\\ 0.566\\ 0.566\\ 0.566\\ 0.566\\ 0.566\\ 0.566\\ 0.566\\ 0.566\\ 0.567\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.5778\\ 0.578\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.598\\ 0$	* O * (WIT * PROMPT * CRLF * SCAN * SELK SBLK1 * SBLK1 * SBLK1 * SBLK1 * SELK1 * SCAN * S	UTPUT A C H CONSOL MUI JMP CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ JNZ RET BLANK CHA MVI LDAX CPI JZ DCR RZ JMP ROUTINE S TS THE FO ROUTINE S TS THE FO ROR HANDL JZ ROUTINE S TS THE FO ROR HANDL CALL JZ ROUTINE C DARD HEX CPI LDAX CPI RC UDAN CALL JZ ROUTINE C DARD HEX CPI LDAX CPI RZ DAD DAD	RLF FOLLOWE ALL OPERATI CRLF ALL OPERATI CRLF VDMOT B,LF VDMOT B,CR VDMOT C. C. SCHR C C. SSBLK1 C. SSBLK1 C. SSBLK1 C. SSPACE C. SSCHR1 CANS OVER C. LLOWING ADD ER. SSCHR1 CANS OVER C. LLOWING ADD ER. SSLK ERR1 ONVERTS ASC CONVERSION. TERED. PACENTIAL SCH CONVERSION. SSCH CONVERTS ASC CONVERSION. SCH CONVERSION. SSCH CONVERSION. TERED. D. D C. SCH CONVERSION. SCH CONVERSION. SCH CONVERTS ASC CONVERSION. TERED. CONVERSION. TERED. CONVERSION. CONVERTS ASC CONVERSION. CONVERTS ASC CONVERSION. CONVERTS ASC CONVERSION. CONVERTS ASC CONVERSION. CONVERTS ASC CONVERSION. CONVERTS ASC CONVERSION. CONVERSION	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO MAKE ROOM FOR THE NEW ONE				
$\begin{array}{c} 0.552\\ 0.553\\ 0.555\\ 0.557\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.577\\ 0.578\\ 0.588\\ 0.590\\ 0.595\\ 0.595\\ 0.593\\ 0.599\\ 0.$	* O * (WIT * PROMPT * CRLF * SCAN * SELK SBLK SBLK1 * SBLK1 * SBLK1 * SCAN * SCAN	UTPUT A C H CONSOL MUI JMP CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ JZ INX CPI JZ JZ JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RNZ JMP ROTINE S TS THE FO CR ROTINE S TS THE FO CALL JZ ROUTINE S TS THE FO CALL JZ ROUTINE C DARD HEX CPI ROUTINE C DARD HEX CPI ROUTINE C DARD HEX CPI ROUTINE C DARD HEX CPI S SECOUNTER ON TH LDAX CPI ROUTINE C DARD HEX CPI ROUTINE C DARD HEX CPI RZ DAD DAD DAD DAD DAD DAD	RLF FOLLOWE ALL OPERATI CRLF B,L'>' VDMOT B,CR VDMOT B,CR VDMOT C C C SELANK SCHR C C SBLK1 C C SBLK1 C C SPACE C C SPACE SPACE C SCHR1 CANS OVER C LLOWING ADD ER. SCHR1 CANS OVER C CONVERSION. D C CONVERSION. D C CONVERSION. D C CONVERSION. D C CONVERSION. C CONVERSION. D C CONVERSION. C CONVERSION. C CONVERSION. C CONVERSION. C C CONVERSION. C C C C C C C C C C C C C C C C C C C	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO MAKE ROOM FOR THE NEW ONE				
$\begin{array}{c} 0.552\\ 0.553\\ 0.555\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.556\\ 0.557\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.599\\ 0.593\\ 0.599\\ 0.$	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK SBLK SBLK SBLK SBLK SBLK SCAN * * SCAN * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * * SCONV * * * * SCONV * * * * SCONV * * * * SCONV * * * SCONV * * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * * * SCONV * * * * SCONV * * * SCONV * * * SCONV * * * * SCONV * * * SCONV * * * SCONV * * * SCONV * * SCONV * * * SCONV * * SCONV * * SCONV	UTPUT A C H CONSOL MUI CALL MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ JZ INX RET DCR JJZ INX RET PAST UP T BLANK CHA MVI LDAX CPI RNZ JMP ROTINE S TS THE FO ADD HEX CALL JZ ROUTINE S TS THE FO CALL CALL CALL DAD DAD DAD DAD CALL	RLF FOLLOWE ALL OPERATI CRLF B,L'>' VDMOT B,CR VDMOT B,CR VDMOT O 12 CHARAC C,12 D BLANK SCHR C C SBLK1 C C SBLK1 C C SPACE C SPACE C SPACE C SCHR1 CANS OVER C LLOWING ADD ER. SCHR1 CANS OVER C LLOWING ADD ER. SBLK ERR1 ONVERTS ASC CONVERSION. P C C SCHR1 CANS OVER C LLOWING ADD C C SCHR1 CANS OVER C LLOWING ADD C C SCHR1 CANS OVER C LLOWING ADD C C SCHR1 CANS OVER C LLOWING ADD C C SCHR1 CANS OVER C C LLOWING ADD C C SCHR1 CANS OVER C C LLOWING ADD C C SCHR1 C CONVERTS ASC CONVERSION. C C CONVERSION. C C C C C C C C C C C C C C C C C C C	D BY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO MAKE ROOM FOR THE NEW ONE DO THE CONVERSION				
$\begin{array}{c} 0.552\\ 0.553\\ 0.555\\ 0.556\\ 0.557\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.578\\ 0.588\\ 0.590\\ 0.593\\ 0.599\\ 0.$	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK SBLK SBLK SBLK SBLK SBLK SCAN * * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * * * * * * * * * * * *	UTPUT A C CALL MVI CALL MVI JMP MVI CALL MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RNZ JNZ RET BLANK CHA MVI LDAX CPI RNZ JMP ROUTINE S TS THE FO CALL JZ ROUTINE S TS THE FO CALL JZ ROUTINE S TS THE FO CALL JZ ROUTINE S TS THE FO DAD DAD DAD DAD DAD CALL JNC ADD CALL JNC CALL JNC CALL JNC CALL JNC CALL CALL CPI RZ CPI RZ CPI CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CALL CPI RZ CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ CALL CPI RZ	RLF FOLLOWE ALL OPERATI CRLF B,LF VDMOT B,CR VDMOT B,CR VDMOT O 12 CHARAC C,12 D BLANK SCHR C C SBLK1 C C SBLK1 C C SPACE C C SCHR1 CANS OVER C LLOWING ADD ER. SELK ER C SBLK ER C CONVERSION. FRED. PAR E SCREEN WI H,D 20H C H H H H H H H H H H H H H H H H H H	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO MAKE ROOM FOR THE NEW ONE				
$\begin{array}{c} 0.552\\ 0.553\\ 0.555\\ 0.556\\ 0.557\\ 0.577\\ 0.578\\ 0.578\\ 0.578\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.588\\ 0.599\\ 0.593\\ 0.599\\ 0.$	* O * (WIT * PROMPT * CRLF * * SCAN * SBLK SBLK SBLK SBLK SBLK SBLK SBLK SCAN * * * SCAN * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCAN * * SCHR SCHR SCONVER * * * THE SE * * * CONVER * SCONV * * * * * * * * * * * * *	UTPUT A C ITPUT A C ALCONSOL CALL MVI JMP MVI CALL MVI CALL MVI DOVER UP T MVI LDAX CPI JIX DCR PAST UP T BLANK CHA MVI LDAX CPI RNZ JMP RET RNZ JMP ROUTINE S ROR HANDL CALL JZ ROUTINE S ROR HANDL CALL JZ ROUTINE S ROR THANDL CALL JZ CPI RIS ENCOUN TER ON TH LDAX CPI RZ JMP ROUTINE S CALL JZ CPI RZ DAD DAD DAD DAD DAD CALL JNC ADD MOV INX	RLF FOLLOWE ALL OPERATI CRLF B,'>' VDMOT B,LF VDMOT C,12 D BLANK SCHR D C SSCHR D C SSELK1 C,10 D SPACE SCHR1 CANS OVER C C LOWING ADD ER. SSCHR1 CANS OVER C C LOWING ADD ER. SELK ERR1 ONVERTS ASC CONVERSION. TERED. PAR E SCREEN WI H,0 D 20H SCHR1 E SCREEN WI H,0 D 20H SCHR1 C SCHR1 C CANS OVER C C LOWING ADD ER. SCHR1 CANS OVER C C LOWING ADD ER. SCREN C CONVERSION. TERED. PAR E SCREEN WI H,0 D C CONVERSION. CONVE	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO MAKE ROOM FOR THE NEW ONE DO THE CONVERSION NOT VALID HEXIDECIMAL VALUE MOVE IT IN BUMP THE POINTER				
$\begin{array}{c} 0.552\\ 0.553\\ 0.553\\ 0.553\\ 0.553\\ 0.555\\ 0.557\\ 0.577\\ 0.$	* O * (WIT * ORLF * CRLF * SCAN * SELK SBLK1 * SELK SBLK1 * * SCAN * A NON * SELK SCHR SCHR1 * THIS * CONVER * THIS * CONVER * THIS * A SCAN * THIS * CONVER * THIS * A SCAN * SCAN * HIS * CONVER * THIS * A SCAN * SCONV * THIS * A SCAN * SCONV * THIS * A SCAN * SCONV * THIS * A SCAN * SPACE * CONVER * A SCAN * SPACE * CONVER * A SCAN * SPACE * CONVER * CONVE	UTPUT A C H CONSOL CALL MVI JMP OVER UP T MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR DCR JZ PAST UP T BLANK CHA MVI LDAX CPI BLANK CHA MVI LDAX CPI RZ JNZ RET PAST UP T BLANK CHA MVI LDAX CPI RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ JNZ RET DCR RZ DCR RZ JNZ RZ DCR RZ JNZ RZ DCR RZ JNZ RZ DCR RZ JNZ RZ DCR RZ JNZ RZ INX DCR RZ JNZ RZ DCR RZ JNZ RZ INX DCR RZ JNZ RZ INX DCR RZ JNZ RZ INX DCR RZ JNZ RZ INX DCR RZ JNZ RZ INX DCR RZ JNZ RZ INX DCR RZ JNZ RZ INX DCR RZ DAD DAD DAD DAD DAD DAD DAD DA	RLF FOLLOWE ALL OPERATI CRLF B,L'>' VDMOT B,LF VDMOT C,12 D BLANK SCHR D C SSCHR D C SSBLK1 C,10 D SPACE SCHR1 CANS OVER C CLLOWING ADD ER. SBLK ERR1 ONVERTS ASCC CONVERSION. TERED. PAR E SCREEN WI H,0 D 20H SCHR1L L,A D SHE1	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR TH A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO MAKE ROOM FOR THE NEW ONE DO THE CONVERSION NOT VALID HEXIDECIMAL VALUE MOVE IT IN BUMP THE POINTER				
03522 05534 05553 05556 05557 05556 05557 05556 05557 05578 05589 05595 05595 05595 05595 05595 05595 05595 05577 05577 05577 05577 05576 05577 05577 05577 05578 05595 05595 05595 05595 05595 05595 05595 05595 05595 05577 05577 05577 05577 05577 05577 05577 05579 05589 05595 05595 05595 05595 05595 05595 05577 05577 05577 05577 05577 05579 05589 05595 05595 05595 05595 05577 05577 05577 05577 05577 05579 05589 05595 05595 05577 05577 05577 05577 05579 05595	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK SBLK SBLK SBLK * * A NON * * CONVER * * CONVER * * * CONVER * * * * CONVER * * * CONVER * * CONVER * * * CONVER * * * CONVER * * CONVER *	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR JZ PAST UP T BLANK CHA MVI LDAX CPI BLANK CHA MVI LDAX CPI RZ DCR RZ JZ RET PAST UP T BLANK CHA MVI LDAX CPI RZ JZ RET DCR RZ JZ RET DCR RZ JZ RET DCR RZ JZ RET DCR RZ JZ RET DCR RZ JZ RET DCR RZ JZ RET DCR RZ JZ RET DCR RZ JZ RET DCR RZ JZ CDA CALL JZ CDAL DAD DAD DAD DAD DAD DAD DAD	RLF FOLLOWE ALL OPERATI CRLF B,L'>' VDMOT B,LF VDMOT C,12 CHARAC C,12 D BLANK SCHR D C SSHK1 C,10 D SPACE SELK1 C,10 D SPACE SCHR1 CANS OVER C CLLOWING ADD ER. SLK ERR1 CONVERTS ASC CONVERSION. TERED. PAR E SCREEN WI H,0 D 20H SHEL PAC SHEL SHE1 L A SHE1 48 10	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RRSS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR IN A SUPPOND HEN AN ASCII MAKE ROOM FOR THE NEW ONE DO THE CONVERSION NOT VALID HEXIDECIMAL VALUE MOVE IT IN BUMP THE POINTER REMOVE ASCII BLAS				
03522 0553 05553 05553 05556 05557 05556 05557 05556 05557 05577 05578 05576 05577 05578 05577 05578 05577 05578 05577 05578 05577 05578 05577 05578 05577 05578 05578 05577 05578 05577 05578 05579 05588 05580 05577 05578 05578 05578 05578 05589 05593 05593 05593 05593 05593 05593 05595 05593 05595 05595 05595 05595 05577 05578 05589 05593 05593 05593 05593 05593 05593 05593 05593 05593 05598 05599 05593 05593 05595 05598 05599 05598 05599 05598 05599 05598 05599 05598 05599 05598 05599 05598 05598 05599 05598 05600 06601 06611 06612 0	* O * (WIT PROMPT * CRLF * * SCAN * SBLK SBLK SBLK SBLK SBLK * * A NON * * CONVER * THIS * CONVER * THIS * CONVER * THIS * A STAN * SPACE * CONVER * * HCONV	UTPUT A C H CONSOL CALL MVI JMP MVI CALL MVI JMP OVER UP T MVI LDAX CPI JZ INX DCR PAST UP T BLANK CHA MVI LDAX CPI BLANK CHA MVI LDAX CPI RZ JZ RZ DCR RZ JZ RZ DCR RZ JZ RZ DCR RZ JZ RZ DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ INX DCR RZ JZ CALL CALL CA	RLF FOLLOWE ALL OPERATI CRLF B,L'>' VDMOT B,LF VDMOT B,CR VDMOT 0 12 CHARAC C,12 D BLANK SCHR D C SSCHR D C SSBLK1 O 10 BLANK RACTER. C,10 D SPACE SCHR1 CANS OVER C LLOWING ADD ER. SBLK ERR1 CONVERTS ASC CONVERSION. TERED. PAR E SCREEN WI H,0 D 20H SHLK ERR1 L L,A D SHE1 48 10	D EY A PROMPT ONS ARE ON THE SCREEN) THE PROMPT PUT IT ON THE SCREEN LINE FEED CARRIAGE RETURN PUT IT OUT AND RETURN TERS LOOKING FOR A BLANK MAXIMUM COMMAND STRING GOT A BLANK NOW SCAN PAST IT NO MORE THAN TWELVE GO BACK WITH ZERO FLAG SET POSITIONS LOOKING FOR SCAN TO FIRST NON BLANK CHR WITHIN 10 GET NEXT CHARACTER WE'RE PAST THEM NEXT SCAN ADDRESS COMMAND ERROR KEEP LOOPING HARACTERS, PAST BLANKS AND RESS TO HEX. ERRORS RETURN TO II DIGITS INTO BINARY FOLLOWING THE SCAN STOPS WHEN AN ASCII AMETER ERRORS REPLACE THE ERROR IN A QUESTION MARK. CLEAR H & L GET CHARACTER IS IT A SPACE? IF SO MAKE ROOM FOR THE NEW ONE DO THE CONVERSION NOT VALID HEXIDECIMAL VALUE MOVE IT IN BUMP THE POINTER REMOVE ASCII BIAS IF LESS THAN 9 IT'S A LITTEP22				

CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC3CIC4DCIC5CIC3CIC5CIC3CIC5CIC3CIC5CIC3CIC5CIC3CIC5CIC3CIC5CIC3CIC5CIC3CIC5CIC3CIC5CIC3CID5CIC3CID6CIC3CID7FECID7FECID7FECID7FECID7FECID7FECID7FECID7CIC3CID7FECID7FECID7FECID7CIC3CID7FECID7</t

C264	C9			062	3		RET		WITH TEST IN HAND
C265 C265				062	4 * 5 *				
C265 C265				062 062	6 * 7 * PC	SYST DINTER	FEM START R, FALLING	UP, CLEAR G THROUGH T	PART OF RAM AND SET STACK O TERMINAL MODE.
C265				062	8 *			,	
C265 C266	AF 4F			062	9 SIR. 0	LA	MOV	C,A	WE CLEAR THE FIRST 256 BYTES
C267 C26A	21	00	C8	063 063	1 2 *		LXI	H,SYSRAM	POINT TO SYSTEM RAM
C26A	77			063	3 CLEF	RA	MOV	M,A	
C26C	0C			063	5		INR	C	
C26D C270	C2	6A	C2	063 063	6 7 *		JNZ	CLERA	CLEAR FIRST 256 BYTES
C270				063	8 *				
C270				063	0 *			TERM COMMA	ND
C270				064	1 * 2 *	THIS	ROUTINE	SETS CHARAC	TERS FROM THE SYSTEM KEYBOARD
C270				064	3 * 1	AND OU	JTPUTS TH	EM TO THE S	ERIAL OUTPUT PORT. IT IS
C270 C270				064	4 * 1 5 * 1	lnteni fermin	NAL. COM	MAND KEYS A	SOL AS A STANDARD VIDEO RE NOT OUTPUT TO THE OUTPUT
C270				064	6 * 1 7 * 7	PORT H	BUT ARE II	NTERPRETED .	AS DIRECT SOL COMMANDS.
C270				064	8 * 5	SoL IN	N THE COM	MAND MODE.	D DI IIII KEIDORKD, IOIO IIII
C270 C270				064 065	9 * 0 *				
C270	31 CD	00	CC	065	1 TERM	vī.	LXI	SP,SYSTP	SET STACK POINTER
C273 C276	CD	C5	C0	065	3		CALL	PERSE	CLEAR THE SCREEN
C279 C279	CD	22	CO	065	4 * 5 KTN		CALL	KSTAT	IS THERE ONE WAITINGT
C27C	C2	90	C2	065	6		JNZ	TIN	
C27F	DB 47	FC		065	8		IN MOV	KDATA B,A	GET THE CHARACTER
C282	E6	80 80	<i>a</i> 2	065	9		ANI	80H	COMMAND KEY?
C287	CD	4B	CO	066	1		CALL	VDMOT	PROCESS IT
C28A C28D	C3	90	C2	066 066	2 3 *		JMP	TIN	
C28D	CD	41	C0	066	4 TOUT	Г	CALL	SDROT	OUTPUT IT TO THE SERIAL PORT
C290	CA	79	C2	066	6		JZ	KIN	LOOP IF NOT
C296 C298	DB E6	F9 7F		066 066	7 8		IN ANI	SDATA 7FH	GET DATA NO HIGH BITS FROM HERE
C29A	47	-	<i>a</i> 0	066	9		MOV	B,A	IT'S OUTPUT FROM 'B'
C29B C29E	CD C3	чв 79	C2	067	1		JMP	KIN	LOOP OVER AND OVER
C2A1 C2A1				067	2 * 3 *				
C2A1				067	4 *				
C2A1 C2A1				067	5 * 6 *		DOWP	COMMAND	
C2A1				067	7 * 8 * 0	THT 1799117	IS ROUTIN	E DUMPS CHA	RACTERS FROM MEMORY TO THE WITH CONSOL ALL OUTPUT COES TO
C2A1				067	9 * 1	THE SC	CREEN).	ALL VALUES	ARE DESPLAYED AS ASCII HEX.
C2A1 C2A1				068 068	0 * 1 * 1	THE CO	OMMAND FO	RM IS AS FO	LLOWS:
C2A1				068	2 *		Dumo odda	v1 oddw2	
C2A1 C2A1				068	4 *		Dump add	ri addrz	
C2A1				068	5 * 6 * (THE	VALUES FI	ROM ADDR1 T	O ADDR2 ARE THEN OUTPUT TO THE DDR1 IS SPECIFIED THEN THE
C2A1				068	7 * T	VALUE	AT THAT	ADDRESS IS	OUTPUT.
C2A1 C2A1	CD	3E	C2	068	8 ° 9 DUME	2	CALL	SCONV	SCAN TO FIRST ADDRESS AND CONVERT IT
C2A4	E5 CD	32	C2	069	0 1		CALL	H	SAVE THE VALUE GET THE NEXT
C2A8	E1		~ ~	069	2		POP	Н	
C2A9 C2AC	CA E5	Β4	C2	069	3 4		J Z PUSH	POVER H	NO SECOND VALUE
C2AD	CD	44	C2	069	5		CALL	SHEX	GET SECOND
C2B0 C2B1	C3	в6	C2	069	7		JMP	NPASS	MIND BENDERS
C2B4 C2B4	54			069	8 * 9 POVI	- R	MOV	D.H	NO SECOND PARAMETER COPY FIRST TO DE
C2B5	5D			070	0		MOV	E,L	
C2B6 C2B7	EВ			070	1 NPAS 2 *	55	XCHG	•	HL HAS START, DE HAS END
C2B7	CD DB	1A FC	C2	070	3 DLOC 4	ΟP	CALL	CRLF	
C2BC	FE	80		070	5		CPI	MODE	MODE KEY' WILL ESCAPE THE DUMP
C2BE C2C1	CA CD	B4 DD	C1 C2	070	6 7		JZ CALL	COMND ADOUT	OUTPUT ADDRESS
C2C4	CD	E5	C2	070	8		CALL	BOUT	ANOTHER SPACE TO KEEP IT PRETTY
C2C9	0E	10		070	9 0 *		MVI	C,10	VALUES PER LINE
C2C9 C2CA	7E C5			071 071	1 DLP1 2	1	MOV PUSH	A,M B	GET THE CHR SAVE VALUE COUNT
C2CB	CD	E2	C2	071	3		CALL	HBOUT	SEND IT OUT WITH A BLANK
C2CE C2D1	D2	02 B4	C1	071	5		JNC	COMND	ALL DONE
C2D4 C2D5	C1 23			071	6 7		POP	В H	VALUES PER LINE
C2D6	0D	~	<i>a</i> 2	071	8		DCR	C DI D1	BUMP THE LINE COUNT
C2D7 C2DA	C2 C3	с9 В7	C2 C2	071 072	9 0		JMP	DTOOD	DO A LFCR BEFORE THE NEXT
C2DD C2DD				072	1 * 2 *	יידיז	PUT HT. MC	HEX 16 RTT	VALUE
C2DD	7~			072	3 *		MOT	D11	- II ETDOW
C2DD C2DE	CD	EA	C2	072	- адот 5	1	CALL	A, A HEOUT	11 11/01
C2E1 C2E2	7D			072	6 7 *		MOV	A,L	THEN L FOLLOWED BY A SPACE
C2E2	CD	EA	C2	072	8 HBOU	JT	CALL	HEOUT	
C2E5 C2E7	06 C3	20 4B	C0	072 073	у BOUT 0	Τ.	MVÍ JMP	в, '' VDMOT	CONSOL PUTS IT ON THE SCREEN
C2EA	15			073	1 * 2 UFOI	יייי	MOV	C A	CPT TUP CUADACTED
C2EB	0F			073	3		RRC	~ /	
C2EC C2ED	0F 0F			073 073	4 5		RRC RRC		MOVE THE HIGH FOUR DOWN
C2EE	0F	Ľ.J	C 2	073	6		RRC	HEOU1	DITT THEM OIT
C2EF C2F2	79	гs	CZ	073	8		MOV	A,C	THIS TIME THE-LOW FOUR
C2F3 C2F3	ЕG	0F		073 074	9 * 0 HEOI	J1	ANI	OFH	FOUR ON THE FLOOR
C2F5	C6	30		074	1		ADI	48	WE WORK WITH ASCII HERE
C2F7 C2F9	гЕ DA	эА FE	C2	074	⊿ 3		JC	OUTH	YUPI
C2FC C2FE	C6 47	07		074	4 5 ОПТТ	H	ADI MOV	7 B,A	MAKE IT A LETTER OUTPUT IT FROM REGISTER 'B'
C2FF	C3	4B	C0	074	6		JMP	VDMOT	
C302 C302				074	/ * 8 *	COMP	ARE DE ANI	D HL	
C302	710			074	9 * 0 acor	ИÞ	MOV	Δ.Τ.	
C303	93			075	1 1		SUB	E	
C304 C305	7C 9A			075 075	∠ 3		MOV SBB	A,H D	
C306	C9			075	4 5 *		RET	•	FLAGS ARE SET
C307				075	6 *				
C307 C307				075	'/* 8*		ENTER	COMMAND	
C307				075	9 * 0 *	THIS	ROUTINE	GETS VALUES	FROM THE KEYBOARD AND ENTERS
C307				076	0 ~ 14 1 * A	STANI	DARD 'GCL	I. IHE INP IN' INPUT S	O ON SCREEN EDITING MAY TAKE
C307				076	2 * PI 3 * די	LACE H	PRIOR TO	THE LINE TE	RMINATOR. A BACK SLASH '/' NS CONTROL TO THE COMMAND MODE
C307	<i></i>	<u>.</u>	a 2	076	4 *			NEIUR	
C307	CD	ЗE	C2	076	5 ENTH	sĸ	CALL	SCONV	SCAN OVER CHARS AND GET ADDRESS

,00		PUSH	н	SAVE ADDRESS
767 768	* ENLOP	CALL	CRLF	
769 770		MVI CALL	B,':' CONT	GET LINE OF INPUT
771 772		CALL MVI	CREM C,1	REMOVE THE CURSOR START SCAN
773 775		CALL XCHG	VDAD2	GET ADDRESS TO DE
776 777	* ENLO1	MVI	C,3	NO MORE THAN THREE SPACES BETWEEN VALUES
000 778		CALL JZ	SCHR1 SCAN ENLOP	TO NEXT VALUE LAST ENTRY FOUND START NEW LINE
779	ENLO2	LDAX	D '/'	GET THE CHR COMMAND TERMINATOR?
81		JZ	COMND	IF SO CONVERT VALUE
83		MOV	A,L	GET LOW PART AS CONVERTED
85 86		MOV	M,A	PUT IN THE VALUE
37		PUSH	H ENIL O1	BACK GOES THE ADDRESS
39	*	OMP	ENDOT	CONTINUE THE SCAN
1	*	EVI	CUTTE COMMAN	ت
3	* * "דעד פ	ROUTTINE	SECTE COMMAN	LOWING DARAMETER AND DORS A
5	* PROGRAM	M JUMP TO	THE LOCATION	DN GIVEN BY IT. IF PROPER
7	* IT CAN *	DO A STAI	NDARD 'RET'U	JRN TO THE CONSOL COMMAND MODE.
99)0	* EXEC	CALL	SCONV	SCAN PAST BLANKS AND GET PARAMETER
)1)2	*	PCHL		GO (AMD TAKE NOTE)
3 4	*			
5 6	*	TAPE 1	LOAD COMMANI	
)7)8	* THIS * THE REA	ROUTINE I AD DATA II	READS FROM E NTO MEMORY.	EITHER TAPE UNIT PLACING WHILE SPACE WITHIN CONSOL
9 0	* DOES NO * COMMANI	OT ALLOW 1 O WILL LOA	FOR 'STANDAF AD Sol- BASI	RD' TAPE ROUTINES THIS IC5 AND OTHER STANDARD SoL
1 2	* SYSTEM *	SOLFTWAR	E FOR DIRECT	F EXECUTION.
. 3 . 4	TLOAD	CALL	SBLK	SCAN TO SPEED PARAMETER
15 16		CALL	SHEX	CONVERT IT
18		ANI	А, L 1 2 20	ONLY BIT ZERO COUNTS
. 7 20	*	JNZ	A, 52 SETSP	EVETEND ITS STOM
22 22	DFLT	XRA	A TADE1	MAKE IT FAST
24	SEISP	OUT	STAPT	START TAPES AND SELECT SPEED
26	*	IN	TDATA	CLEAR THE UART FLAGS
8	TLOD1	CALL	RHEAD	READ PAST HEADER
30	*	THID	BLOCK	GET BLOCK SIZE
32		XCHG LHLD	LOADR	TO DE GET LOAD ADDRESS
84 85	* LOLOOP	MOV	A.D	GET COUNT
6 7		ORA JZ	E TOFF	COUNT IS ZERO-TURN OFF TAPE AND RETURN
38 39		LXI XCHG	в,-256	THIS MANY PRIOR TO CRC TEST COUNT TO HL
40 41		DAD JNC	B LBLK	A LITTLE MATH NO CARRY, IT'S THE LAST BLOCK
2	*	MVI	в,0	256 TO READ
14 15	RDBLK	MVI XCHG	C,0	ZERO THE CRC ROUND ROBIN
00 16	* RTBYT	CALL	TAPIN	GET CHARACTER
7 8		MOV INX	М,А Н	STORE IT BUMP MEMORY LOCATION
19 50		XRA CMA	C	UPDATE THE CRC
51 52		SUB MOV	C C,A	STORE THE NEW
53 54		JNZ	B RTBYT	COUNT DOWN STILL MORE IF NOT ZERO
56 56		CALL	CRCCK	CHECK CRC AND FALL THROUGH TO ERROR IF NO GOOD
58	* סקקיי	U Z MVT	B IGI-40TT	IESI UK
50 51	TELK	CALL	VDMOT	PUT IT ON THE SCREEN
52 52	*	UMP	COMIND	
54	* CONS	SOL ERROR	HANDLER	
56	ERR1 ERR2	XCHG MVT	M.'?'	GET SCAN ADDRESS PUT A QUESTION MARK THERE
58	*	JMP	COMND	AND GO TO COMMAND MODE
70 71	* READ *	THE HEAD	DER	
72 73	RHEAD RHEA1	MVI IN	B,10 STAPT	FIND 10 NULLS GET A BYTE
74 75	-mont	ANI JZ	TDR RHEA1	
76 77		IN ORA	TDATA A	IGNORE ERROR CONDITIONS ZERO?
78 79		JNZ DCR	- RHEAD B	
30 31	*	JNZ	RHEA1	LOOP UNTIL 10 IN A ROW
32 33	* WAI1	r for the	START CHARA	ACTER
84 85	SOHL	CALL DCR	TAPIN A	
-	*	JNZ	SOHL	WAIT FOR A '1'
36 37	* NOW	GET THE	HEADER	
86 87 88 89	^	LXI	H, THEAD B, HLEN*256	POINT TO BUFFER LENGTH OF HEADER IN 'B',C <o< td=""></o<>
86 87 88 89 90 91	^	LXI	,	
36 37 38 39 90 91 92 93	* RHED1	LXI CALL	TAPIN	GEI DIIE
36 37 38 90 91 92 93 94 95	* RHED1	LXI CALL MOV INX	TAPIN M,A H	STORE IT INCREMENT ADDRESS
36 37 38 90 91 92 94 96 97	* RHED1	LXI CALL MOV INX XRA CMA	TAPIN M,A H C	STORE IT INCREMENT ADDRESS NOW CALCULATE THE CRC INSIDE OUT AND UPSIDE DOWN
167 189 101 123 145 167 189	* RHED1	LXI CALL MOV INX XRA CMA SUB MOV	TAPIN M,A H C C C,A	SUI BILE STORE IT INCREMENT ADDRESS NOW CALCULATE THE CRC INSIDE OUT AND UPSIDE DOWN SQUEEZE IT AND SAVE AGAIN
36 37 38 39 11 12 34 35 67 18 39 10 11 12 13 14 15 16 7 18 9 10 11	* RHED1	LXI CALL MOV INX XRA CMA SUB MOV DCR JNZ	TAPIN M,A H C C C,A B RHED1	SEI BILE STORE IT INCREMENT ADDRESS NOW CALCULATE THE CRC INSIDE OUT AND UPSIDE DOWN SQUEEZE IT AND SAVE AGAIN WHOLE HEADER YET? LOOP UNTL DONE
678901234567890123	* RHED1	LXI CALL MOV INX XRA CMA SUB MOV DCR JNZ ROUTINF	TAPIN M,A H C C,A B RHED1 GETS THE NF	GLI BILL STORE IT INCREMENT ADDRESS NOW CALCULATE THE CRC INSIDE OUT AND UPSIDE DOWN SQUEEZE IT AND SAVE AGAIN WHOLE HEADER YET? LOOP UNTIL DONE SXT BYTE AND COMPARES IT
67890123456789012345	* THIS * THIS * TO THE * RETURN	LXI CALL MOV INX XRA CMA SUB MOV DCR JNZ ROUTINE VALUE IN	TAPIN M,A H C C C,A B RHED1 GETS THE NH N REGISTER (GET BILE STORE IT INCREMENT ADDRESS NOW CALCULATE THE CRC INSIDE OUT AND UPSIDE DOWN SQUEEZE IT AND SAVE GGAIN WHOLE HEADER YET? LOOP UNTIL DONE EXT BYTE AND COMPARES IT . THE FLAGS ARE SET ON
6789012345678901234567	* THIS * THIS * TO THE * RETURN * CRCCK	LXI CALL MOV INX XRA CMA SUB MOV DCR JNZ ROUTINE VALUE IN CALL	TAPIN M,A H C C C C,A B RHED1 GETS THE NE N REGISTER (TAPIN	GET EFILE STORE IT INCREMENT ADDRESS NOW CALCULATE THE CRC INSIDE OUT AND UPSIDE DOWN SQUEEZE IT AND SAVE AGAIN WHOLE HEADER YET? LOOP UNTIL DONE EXT BYTE AND COMPARES IT C. THE FLAGS ARE SET ON GET CRC BYTE

C300E5C300C01AC2C301CDC1C313CD3DC1C316CD25C1C316CD34C2C317CD34C2C321CA64C1C322CA64C1C323CA64C1C324CAC3C3C325CA64C1C326CAC3C3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C335CACAC3C336CACAC3C337CDCAC3C347CAC3C3C348CAC3C355CACAC3C347CAC3C347CAC3C347CAC3C347CAC3C346CAC3C357CACAC356CACAC357</

0909 0910	*	RET		
0911 0912	* * THIS	3 ROUTINE	GETS THE NI	EXT AVAILABLE BYTE FROM THE
0913 0914	* TAPE. * FOR A	WHILE W	AITING FOR S	THE BYTE THE KEYBOARD IS TESTED RECEIVED THE TAPE LOAD IS
0915	* TERMIN	NATED AND	A RETURN TO	O THE COMMAND MODE IS MADE.
0917 0918	TAPIN	IN ANI	STAPT TDR	CHECK STATUS
0919 0920		JNZ TN	TREDY	ONE IS AVAILABLE CHECK FOR MODE WHILE WE'RE WAITIN
0921		CPI	MODE	MODE WAS CIVEN ADOPT OPEDATION
0923		JMP	TAPIN	NOT MODESTAY IN LOOP
0924 0925	TREDY	IN	STAPT	
)926 0927		ANI JNZ	TFE+TOE TERR	DATA ERROR? IF FRAMING OR OVERRUN ERROR
)928 0929		IN RET	TDATA	GET THE DATA
)930 0931	* * THIS RO	OUTINE CAL	LCULATES TH	E LENGTH OF THE LAST BLOCK
)932 0933	* LBLK	LXI	B1	
0934		DAD TNX	В Н	COMPLEMENT HL TWO'S
0936		MOV	B,L H O	LENGTH TO REGISTER 3
0938	*	JMP	RDBLK	ONWARD TO THE END
0940	* THIS	ROUTINE	TURNS THE	TAPE UNITS OFF
0941	TOFF	XRA	A	
0943 0944		OUT RET	STAPT •	GIVE COMMAND AND GRIND TO A SLOW STOP
0945 0946	*			
)947 0948	DELAY DLOP1	LXI DCX	D,0 D	START LOOP DOWN COUNT
)949 0950		MOV ORA	A,D E	TEST FOR ZERO
0951 0952		JNZ RET	DLOP1	IF NOT
0953	*			
0955 0954	*			
0957 0957	*		<< SoL SYS	STEM EQUATES >>
0959	*			
0960 0961	*	VDM PAI	RAMETERS	
)962 0963	VDMEM *	EQU	0CC00H	VDM SCREEN MEMORY
)964 0965	*	KEYBOAI	RD SPECIAL I	XEY ASSIGNMENTS
)966 0967	* DOWN	EOU	9AH	
0968	UP	EQU	97H 81H	
0970	RIGHT	EQU	93H	TOND KEY
0972	MODE	EQU	80H	LOAD KEI
0974	HOME	EQU	08EH	
0976	BACKS LF	EQU	5FH 10	BACKSPACE
)977 0978	CR BLANK	EQU EQU	13	
)979 0980	SPACE CX	EQU EQU	BLANK 'X'-40H	
)981 0982	*	PORT A	SSIGNMENTS	
0983	* STAPT	EOU	OFAH	STATUS PORT GENERAL
0985	SERST	EQU	OF8H OF9H	SERIAL STATUS PORT
0987	TDATA	EQU	OFBH	TAPE DATA
0989	PDATA	EQU	0FDH	PARALLEL DATA
0991	SENSE	EQU	OFFH	SENSE SWITCHES
0992	*			
)994 0995	*	BIT AS	SIGNMENT MA	SKS
J996 0997	* SCD	EQU	1	SERIAL CARRIER DETECT
)998 0999	SDSR SPE	EQU EQU	2 4	SERIAL DATA SET READY SERIAL PARITY ERROR
1000 1001	SFE SOE	EQU EQU	8 16	SERIAL FRAMING ERROR SERIAL OVERRUN ERROR
1002	SCTS SDR	EQU EOU	32 64	SERIAL CLEAR TO SEND SERIAL DATA READY
1004	STBE *	EQU	128	SERIAL TRANSMITTER BUFFER EMPTY
1006	KDR DDP	EQU	1	KEYBOARD DATA READY
1008	PXDR	EQU	- 4 8	PARALLEL DEVICE READY
1010	TOE	EQU	16	TAPE OVERFLOW ERROR
1011	TDR TTBE	EQU EQU	128	TAPE DATA READY TAPE TRANSMITTER BUFFER EMPTY
1013	SOK	EQU	1	SCROLL OK FLAG
1015 1016	* TAPE1	EQU	64	TAPE ONE 'ON' BIT
1017 1018	TAPE2 *	EQU	128	TAPE TWO
1019	*			
1021	*	S	DL SYSTEM GI	LOBAL AREA
1023	*	ORG	008004	START OF 1K RAM APFA
1025	*	FOU	6	CTADE OF CUCTEM DAM
1025	SYSTP	EQU	\$+1024	STACK IS AT THE TOP
⊥∪28 1029	*			
	*	CONSO	L PARAMETER	AREA
1030 1031	*		1	CURRENT CHARACTER POSITION
1030 1031 1032 1033	* NCHAR LINE	DS DS	1	CORRENT LINE POSITION
1030 1031 1032 1033 1034 1035	* NCHAR LINE BOT OPORT	DS DS DS DS	1 1 1	BEGINNING OF TEXT DISPLACEMENT OUTPUT PORT
1030 1031 1032 1033 1034 1035 1036 1037	* NCHAR LINE BOT OPORT IPORT *	DS DS DS DS DS	1 1 1 1	BEGINNING OF TEXT DISPLACEMENT OUTPUT PORT INPUT PORT
1030 1031 1032 1033 1034 1035 1036 1037 1038	* NCHAR LINE BOT OPORT IPORT * *	DS DS DS DS DS	1 1 1 1	DEGINITION OF TEXT DISPLACEMENT OUTPUT PORT INPUT PORT
1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040	* NCHAR LINE BOT OPORT IPORT * * * THEAD	DS DS DS DS DS DS	1 1 1 1	NAME
1030 1031 1032 1033 1034 1035 1035 1036 1037 1038 1039 1040 1041 1042	* NCHAR LINE BOT OPORT IPORT * * * THEAD	DS DS DS DS DS DS DS DS DS	1 1 1 1 1 5 1 1	NAME THIS BYTE MUST BE ZERO
1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044	* * NCHAR LINE BOT OPORT IPORT * * * THEAD HTYPE BLOCK LOADR	DS DS DS DS DS DS DS DS DS DS DS DS	5 1 2 2 2	NAME THIS BYTE MUST BE ZERO TYPE BLOCK SIZE LOAD ADDRESS
1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046	* * NCHAR LINE BOT OPORT IPORT * * THEAD HTYPE BLOCK LOADR XEQAD HSPR	DS DS DS DS DS DS DS DS DS DS DS DS DS D	1 1 1 1 1 1 2 2 2 3	NAME THIS BYTE MUST BE ZERO TYPE LOAD ADDRESS AUTO EXECUTE ADDRESS SPARES
1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1045 1047 1048	* * NCHAR LINE BOT OPORT IPORT * * THEAD HTYPE BLOCK LOADR KEQAD HSPR * HLEN	DS DS DS DS DS DS DS DS DS DS DS DS DS D	1 1 1 1 1 1 2 2 2 3 3 \$-THEAD	NAME THIS BYTE MUST BE ZERO TYPE BLOCK SIZE LOAD ADDRESS AUTO EXECUTE ADDRESS SPARES LENGTH OF HEADER

ACOMP	2000	ADUUI	ປປມມ	AINP	CUID	AUUI	0007
BACKS	005F	BGIN	C001	BLANK	0020	BLOCK	C80C
BOT	C802	BOUT	C2E5	CHAR	C07B	CLEAR	008B
CLERA	C26A	CLIN1	COE9	CLINE	COE3	COMND	C1B4
COMTA	C194	CONT	C1CC	COPRC	C1E1	CR	000D
CRCCK	C3C1	CREM	C13D	CRLF	C21A	CRPRC	C1DD
CUR	C0C1	CX	0018	DELAY	C3F1	DFLT	C34A
DISPT	C071	DLOOP	C2B7	DLOP1	C3F4	DLP1	C2C9
DOWN	009A	DSTAT	OOFE	DUMP	C2A1	ENLO1	C31C
ENLO2	C324	ENLOP	C30B	ENTER	C307	ERAS1	COCA
ERAS2	COD6	ERAS3	CODD	ERR1	C38E	ERR2	C38F
ERROT	C1AD	EXEC	C335	FDCOM	C1F4	GCLIN	C1C3
GOBAC	C056	GOBK	C05D	HBOUT	C2E2	HCONV	C24B
HCOV1	C25B	HEOU1	C2F3	HEOUT	C2EA	HLEN	0010
HOME	008E	HSPR	C812	HTYPE	C80B	IPORT	C804
ITAB	C18C	KDATA	00FC	KDR	0001	KIN	C279
KREA1	CO27	KREAD	C028	KSTAT	C022	LBLK	C3E1
LEFT	0081	LF	A000	LINE	C801	LOADK	008C
LOADR	C80E	LOLOO	C361	MODE	0080	NCHAR	C800
NCOM	C20B	NEXT	C076	NPASS	C2B6	OCHAR	C081
OK	COAD	OPORT	C803	OTAB	C184	OUTH	C2FE
PBACK	C145	PCR	C14E	PCUR	C113	PDATA	00FD
PDOWN	COFA	PDR	0002	PERSE	C0C5	PHOME	COF3
PLEFT	Clod	PLF	C155	POVER	C2B4	PRIT	C117
PROCS	C1D2	PROMP	C212	PUP	C104	PXDR	0004
RDBLK	C370	RHEA1	C396	RHEAD	C394	RHED1	C3B4
RIGHT	0093	RTBYT	C373	SBLK	C224	SBLK1	C226
SC	C161	SCD	0001	SCHR	C232	SCHR1	C234
SCONV	C23E	SCROL	C098	SCTS	0020	SDATA	00F9
SDR	0040	SDROT	C041	SDSR	0002	SENSE	OOFF
SEROT	C040	SERST	00F8	SETSP	C34B	SFE	0008
SHE1	C247	SHEX	C244	SINP	C013	SOE	0010
SOHL	C3A7	SOK	0001	SOUT	C004	SPACE	0020
SPE	0004	SREA1	C036	SREAD	C037	SROL	C09C
SSTAT	C031	SIAPT	OOFA	STBE	0080	STRTA	C265
SYSRA	C800	SYSTP	CC00	TAPE1	0040	TAPE2	0080
TAPIN	C3C6	TBL	C165	TDATA	OOFB	TDR	0040
TERM	C270	TERR	C386	TFE	0008	THEAD	C805
TIN	C290	TLOAD	C339	TLOD1	C354	TOE	0010
TOFF	C3ED	TOUT	C28D	TREDY	C3D7	TSRCH	CU62
TTBL	0080	UP	0097	VDAD	C128	VDAD2	C125
VDADD	C121	VDMEM	CCUO	VDM01	C04C	VDMOT	CU4B
XEQAD	C810						

C3C5C3C3C6C3C3C6C3C3C6C3C3C6C3C3C6C3C3C7C3C3C7C3C3D7DBC3D7C3C3D7DBC3D7DBC3D7C3C3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7DBC3D7C3C3D7C3C3D2D3C3E1OC3E2C3C3E3C3C3E4D3C3E5C3C3E7C3C3E7C3C3F8C3C3F8<t

(conti	nue	d fr	om pag	e 4)				
009E	22	F1	04	0710		SHLD	SHED+3	5DH FIREMAN
00A1	2A	78	01	0720		LHLD	FRA1	THE LITTLE ENGINE
00A4 00b7	∠∠ 2∆	24 71	05	0730		SHLD I.HI.D	SHED+31 FRA2	AUH ALSO HAD A COWCATCHER
00A7	22	26	05	0750		SHLD	SHED+32	A2H AND LOTS OF FUNNY
00AD	22	28	05	0760		SHLD	SHED+32	A4H WHEELS AND THE
00B0	22	2A	05	0770		SHLD	SHED+32	A6H THINGS THAT CONNECTED
00B3	2A 22	7C	01	0780		LHLD	FRA3	AND TWO VERY TINY
00B0 00B9	22 2A	ZC 7E	01	0780		I'HI'D	FRA4	VERY BACK
00BC	22	2E	05	0810		SHLD	SHED+32	AAH ALTOGETHER IT LOOKED
00BF	2A	80	01	0820		LHLD	FRA5	QUITE LONELY AND YET
00C2	22	30	05 एए	0830		SHLD	SHED+32	ACH IT APPEARED VERY
0000	21	62	05	8850		LXI	H.SHED	+3DEH THERE ON THE
00CB	36	19		0860	RAILS	MVI	м,19н	RAILS
00CD	03			0870		INX	В	WITH NOTHING AT ALL
00CE	23 25			0880		INX VRA	H Z	TO DO
00D0	A8			0900		XRA	B	WELL!
00D1	C2	СВ	00	0910		JNZ	RAILS	LET'S RUN IT JUST FOR FUN
00D4	21	00	CC	0920	HOSTL	LXI	H,RRY	GET TRAIN OUT OF SHED
00D7 8000	ЕВ 21	84	01	0930		XCHG LXT	H.SHED	MOVE IT FROM SHED
00DB	7E	01	01	0950	FIRE	MOV	A,M	LIGHT FIRE
00DC	23			0960		INX	H	
00DD	EB			0970		XCHG	N/T 75	MOVELTE
00DE 00DF	23			0980		TNX	м,А Н	MOVE II
00E0	7C			1000		MOV	A,H	
00E1	EΒ			1010		XCHG		
00E2	FE	D0	0.0	1020		CPI	0D0H ETDE	END OF YARD?
00E4 00E7	C2 CD	БD	00	1040		CALL	TRAVL	GO TAKE TRIP
00EA	C3	D4	00	1050		JMP	HOSTLE	NO ROUND TRIPS, JUST DO AGAIN
00ED				1060	•			
00ED		40	03	1090	TRAVL	LXI	B,64.1.	3 MAKE TRIP 13 MILES
00F0 00F1	CD	FF	00	1000	CIIOO	CALL	STROK	MAKE FORWARD MOTION
00F4	CD	0F	01	1100		CALL	TURN	TURN WHEELS
00F7	C1			1110		POP	B	
0019	0B 0B			1130		DCX XRA	B A	CLICK ODOMETER
00FA	A8			1140		XRA	B	13 MILES YET?
00FB	C2	FO	00	1150		JNZ	CHOO	NO
00FE	C9			1160		RET	•	DO NEXT TRIP
00FF 00FF	21	01	CC	1180	STROK	LXI	н,0СС0	1H MOVE DOWN THE TRACK
0102	0E	D0		1190		MVI	C,ODOH	
0104	7E			1200	COAL	MOV	A,M	ADD MORE COAL
0105	2B 77			1210		DCX	Н	
0100	23			1220		INX	H,A	
0108	23			1240		INX	H	
0109	7C			1250		MOV	A,H	
010A 010B	В9 С2	04	01	1260		TNZ	COAL	NO PUT MORE ON!
010E	C9	01	01	1280		RET	COLL	
010F		~ ~		1290	·			
010F	21	00	CC	1300 1310	TURN	LXI MOV	H, UCCU	JH TURN WHEELS
0112	23			1320	птор	INX	А,М Н	
0114	FE	06		1330		CPI	6	FIND COWCATCHER
0116	C2	12	01	1340		JNZ	HISS	LOOK AGAIN
0119 011a	⊿3 7⋤			1350	FOUND AXI.F	INX MOV	н дм	
011B	FE	11		1370	111111	CPI	11H	WHAT QUARTER TURN?
011D	C2	31	01	1380		JNZ	NXT1	
0120	ΤT	7Ε	14	1400		LXI	D,147EH	H NEXT QUARTER TURN
0123	06	07		1410	AXL1	MVI	в,7	DO 4 AXLES
0125	72			1420	AXL2	MOV	M,D	WHEELS MOVED HERE
0126	05	F 0	0.1	1430		DCR	B	LAST WHEEL?
0127 012A	23	52	01	1440 1450		J Z TNX	WORK H	ON THE RAILROAD
012B	73			1460		MOV	м,Е	(SIDE RODS MOVED HERE)
012C	23			1470		INX	H	ALL THE LIVE
012D	05	25	01	1480		DCR .TN7	Β ΔΥΓΩ	LONG DAY
0131	CΔ	40	ÛŢ	1500			АЛЦИ	
0131	FE	14		1510	NXT1	CPI	14H	QUARTER TURN
0133	C2	3C	01	1520		JNZ	NXT2	T
0139	⊥⊥ C3	⊿⊔ 23	13 01	⊥530 1540		цат JMP	ע∠נו, u, ב AXL1	1
013C	FE	13		1550	NXT2	CPI	13H	QUARTER TURN

013E	C2	47	01	1560		JNZ	NXT3	
0141	11	5F	12	1570		LXI	D,125FH	H
0144	C3	23	01	1580		JMP	AXL1	
0147	FΕ	12		1590	NXT3	CPI	12H	OUARTER TURN
0149	C2	1 A	01	1600		TNZ	AXLE	2
014C	11	20	11	1610		T.XT	112 ח זח 112 ח	4
0145	C 3	22	01	1620		TMD	$\lambda XI.1$	
	00	2.5	01	1620		UME	АЛЦТ	
0152	01	пΟ	ΔD	1640		тут		
	0T 0T	во	UВ	1040	WORK		в,3000	
0155	0B	~ ~		1650	OIL	DCX	в,	
0156	DB	00		1660			0	DID THE CONDUCTOR
0128	E6	40		T0./0		ANI	RDA	FLAG US DOWN?
015A	C2	63	01	1680		JNZ	QUIT	QUITTIN' TIME
015D	AF			1690		XRA	A	NO, THEN HIGHBALL
015E	A8			1700		XRA	В	
015F	C2	55	01	1710		JNZ	OIL	NO SQUEEKS, PLEASE!
0162	C9			1720		RET		~
0163				1730				
0163				1740	. PUT A	JUMP TO	O WHEREN	VER YOU NEED TO RETURN IN THE NEXT
0163	CZ	60	ΕO	1750	OULT	TMP	ALSS	YOUR RETURN LINK COES HERE
0166	05	00	ЦО	1760	2011	0111	11100	
0166				1770	י גחם	FOIT	/∩u	ערגים גייגר סוו∧ר עדסד
0166				1700	KDA DDV	EQU		FOI TOOK DATA KEADI FLAG HERE
0100				1700	CMOVE	EQU	OCCOUR CEU	
0100	01	10		1000	SMOKE	ЕQU	0FH 100111	SMOKE CHARACIER
0166	10	10		1800	CABI	DW	IUUIH	CAB DESCRIPTION
0168	10	5D		1810	CABZ	DW	5DIUH	
016A	7E	20		1820	CAB3	DW	207EH	" "
016C	28	0A		1830	BOI1	DW	0A28H	BOILER DESCRIPTION
016E	0A	19		1840	BOI2	DW	190AH	п п
0170	0A	0A		1850	BOI3	DW	0A0AH	п п
0172	0A	5B		1860	BOI4	DW	5B0AH	II II
0174	39	39		1870	BOI5	DW	3939н	II II
0176	5D	20		1880	BOIG	DW	205DH	II II
0178	06	20		1890	FRA1	DW	2006н	FRAME DESCRIPTION
017A	11	2D		1900	FRA2	DW	2D11H	пп
017C	11	20		1910	FRA3	DW	2011H	пп
017E	6F	2D		1920	FRA4	DW	2D6FH	пп
0180	6F	20		1930	FRA5	DW	206FH	п п
0182	19	19		1940	TIES	DW	1919H '	TTES DESCRIPTION
0184	00			1950	SHED	NOP		THIS IS SHED AREA
0185	00			1960	CLOUD	FOII	• • •	H BEGINNING OF CLOUD
0185				1970	CHOOD	шQU	φ·01/21	I DEGIMINING OF CEOD
0105				1970	•			
0100								
•1000	000	יע ט ו	202021	8401	01000836	20230871	77802091	1021 270211000060207
·1700) 1 7 (500011	20401		20230BAI		50001134000604CD5063
•1A00) 7 A (2000DCD300			
• 1 A O O	,34()/〒(2001026	560020 560020		19017601	1261601	1135000001CD3000C39A N00/260701ND0/266FF7
• 1AU) <u>-</u> 上し	1025	001936			JZIA0U4	TZOTOZI	

:1A0068002A660122AE042A680122B0042A6A0122B2042A6C0122E5042A6E09 :1A0082000122E7042A700122E90422EB042A720122ED042A740122EF042A0D :1A009C00760122F1042A78012224052A7A01222605222805222A052A7C0195 :1A00B600222C052A7E01222E052A800122300501DFFF21620536190323AF52 :1A00D00A8C2CB002100CCEB2184017E23EB77237CEBFED0C2DB00CDED00B1 :1A00EA00C3D400014003C5CDFF00CD0F01C10BAFA8C2F000C92101CC0ED049 :1A0104007E2B7723237CB9C20401C92100CC7E23FE06C21201237EFE11C2DD :1A011E003101117E1406077205CA520123732305C22501FE14C23C01112D5C :1A01380013C32301FE13C24701115F12C32301FE12C21A01112D11C323010C :1A01520001B80B0BDB00E640C26301AFA8C25501C9C300060110105D7E2046 :1A016C00280A0A190A0A0A5B39395D200620112D1206F2D6F20191900CB

:00