volume 2

Disassembled Handbook for TRS-80

Richcraft Engineering Ltd.
Drawer 1065
Chautauqua, New York 14722

- ABOUT THE AUTHOR -

Robert M. Richardson is a graduate of the Mercersburg Academy and a member of the class of '48 at Princeton University. During World War II he was a USAAF Aviation Cadet and later a P-51 Mustang fighter pilot. He was one of the first USAAF jet fighter pilots.

He served as a radiological engineer at Eniwetok Atoll in 1950 and 1951 during a number of atomic tests and the first "H" bomb test which was a spectacular success.

He is a former flight Captain for United Air Lines, was co-founder of Litton Industries' commercial inertial navigation division, President of ITT Navigator Systems a division of International Telephone & Telegraph Corp., and later, World-wide Product Line Executive for ITT World Hdqs. commercial aviation and marine electronics activities.

Mr. Richardson is a prolific author on the subjects of aviation electronics, microwave engineering and computer technology. His latest books include: The Gunnplexer Cookbook - A 10 GHz Microwave Primer, Disassembled Handbook For TRS-80 - Volume 1, and Disassembled Handbook For TRS-80 - Volume 2. He has been a member of the Institute of Electrical & Electronic Engineers since 1952, and a Senior Member since 1960. He resides at Lake Chautauqua, New York where he alternates seasons between racing his "C" scow sailboat and snowplowing a 1/2 mile driveway to his home on the lake front.

TYPOS-VOLUME 2 "DISASSEMBLED HANDBOOK FOR TRS-80" 1ST PRINTING

PAGE 11 - decimal 16 should read: DJNZ <256

PAGE 12 - decimal 128 should read: ADD A,B

decimal 168 should read: XOR B

PAGE 16 - change following decimal instructions:

216 = 3,B 224 = 4,B 225 = 4,C 232 = 5,B 233 = 5,C

234 = 5,D 240 = 6,B 241 = 6,C 242 = 6,D 243 = 6,E

248 = 7,B 249 = 7,C 250 = 7,D 251 = 7,E 252 = 7,H

NOTE (our Kamikaze typist did not bail-out of this one)

PAGE 18 - add following: 54 n/a LD (IX+zz),N

PAGE 19 - decimal 73: OUT (C),C decimal 105: OUT (C),L

PAGE 20 - add following: 54 n/a LD (IY+zz),N

PAGE 87 - 1st line 30 ending: PRINTB-C (not PRINT B*C)

PAGE 143 - line 6: change LID to MID (LID = ham radio saying)

PAGE 150 - 106 [= up-arrow: 109 = 11222 *** OUR APOLOGIES ***

- VOLUME II -

CHAPTER	CONTENTS	PAGE
i	- Foreword	3
ii	- Introduction	4
1.	- Disassembling TRS-80 ROM Object Code	8
2.	- Partial Disassembled Level II ROM	21
3.	- Using VARPTR and ROM Calls With Integers	84
4.	- String & String Array ID, Location, & Recall	91
5.	- Single & Double Precision Floating Point Numbers	95
6.	- Simulating 'JKL' LPRINT Video For Non-Disk Systems	111
7.	- Storing Video IN MEM For Later Recall	115
8.	- Split Screen Separate Video On One Display	119
9.	- Level II ROM CALL & Ancillary Functions Summary	138
10.	- Basic Review: Morse Code Transmit & Receive Pgm.	141
11.	- Basic Review: TV Satellite Locator Program	157
12.	- Self-Test Questions Chapters 1 - 11	162
13.	- Bibliography and Self-Test Answers	166
14.	- APPENDIX: Disk/Cassette Programs - Vols. 1 and 2	171
	Index For Volumes 1 and 2	172

- NOTICE -

TRS-80 is a trademark of the Tandy Corporation. The users of the CALLed Level II ROM subroutines presented herein, must of necessity have purchased Level II BASIC and thereby paid the royalties due the copyright owners. All data in this handbook is purely instructional in nature and serves as a supplement to the original documentation provided by Radio Shack/Microsoft and Zilog. Richcraft's absolute refusal to provide copies or extracts of the original documentation assures the copyright owners that all users must purchase Level II ROM from Radio Shack, thereby paying royalties due. To assure the copyright owners complete protection, approximately 256 object & 256 source codes have been ENTIRELY blanked out in the partial ROM listing provided for reference. Furthermore, a considerable number of Level II ROM disassembled Opcodes and Operands are erroneously printed out by the disassembler. They have not been corrected to protect the copyrighted program.

First Printing: May 1980

Published by: Richcraft Engineering Ltd.
Drawer 1065, Wahmeda Industrial Park
Chautauqua, New York 14772
(716) 753 2654

- FOREWORD -

VOLUME II of the handbook has grown like "Topsy," and is largely a collection of essays and programs that were written to specifically expand on a number of points that were either glossed over or skipped entirely in Volume I. A number of the Chapters attempt to answer specific questions raised by readers of the first volume, largely via developing programs written in either BASIC or assembly language to illustrate the points raised.

The reader of Volume II is expected to have fully mastered the information presented in Volume I and most importantly, is expected to follow the self-programmed learning frame-work in which all concepts and chapters are presented; i.e., DO NOT proceed to the next chapter until you FULLY UNDERSTAND each and every point covered in the chapter under study. If you choose to ignore this advice, you do so at your own peril and ignorance. Volume I's editor jokingly admonished the reader to: "If all else fails, try putting the book under your pillow and sleeping on it." Sad to say, osmosis will not permeate a sleeping intellect no matter how bright or advanced. A goodly number of readers of Volume I commented that the questions and answers sections were extremely helpful in clarifying obscure points, so do not overlook this approach if you are still wondering "what did he say," after re-reading a chapter's contents.

Simply reading a given Chapter's contents and programs is about as worthless as mammary glands on a boar hog, UNLESS you take the time and effort to sit down at your TRS-80 and actually ENTER and use the programs presented to illustrate the points being covered. There is indeed a certain intellectual MAGIC that transpires within the human brain when one physically sits down at the TRS-80, enters the program, runs it, and experiments with it, that simply reading the text will not bring into play. If you choose to ignore this advice and assignment, the only one you are cheating is yourself.

DEDICATION:
Without the generous aid and assistance of Margaret C. Merz and Nancy A. Courtney, neither Volume I or II of this handbook would ever have come to pass. Their continued encouragement and hard work are gratefully acknowledged and appreciated.

ADDITIONAL THANK YOUS:
To the late Charles Tandy for his courage and investment in making the TRS-80 a happening; to Western I/O for an IBM Selectric printer that never quits; to Apparat for NEWDOS 80; to Shrayer for Electric Pencil; to Microsoft's Paul Allen and Bill Gates for Level II BASIC; to Mumford Micro for a 3 speed clock; and last but not least to Radio Shack for building the world's most cost effective and widely used microcomputer. Thank you each and every one.

- INTRODUCTION -

Disassembled source and object code without comments is like a ship without a rudder or a kite without a tail. There it is, but we have not the slightest idea as to WHERE it is going.

Even our disassembler cannot discern the difference between constants, address table entry points and data lists which it disassembles and prints out as utter meaningless/misleading garbage. Our favorite case in point is the Opcode and Operand for EX.....AF,AF' which the disassembler dutifully prints out at numerous locations including: 005AH, 1479H, 1619H, 18BAH, 18BCH, and so forth. In actuality, the alternate register pairs are NEVER used in Level II ROM, which is a decided plus for us duffer TRS-80 programmers. We may use them any time it is convenient without malice aforethought and unknowingly foul-up some important CALL function's operation or other. Conversely, just how do we ever hope to understand how the Level II ROM functions we are CALLing work if we cannot trust what the disassembler is telling us?

We are glad you asked that question, Gridley, as that is what Volume II of this handbook is all about. Though by no means a complete anthology and discourse on each and every Level II ROM function available to us we will cover in considerable depth the ways and means with which integers, single precison/floating point, double precision/floating point, strings and string-arrays are handled and stored in memory. Let's take a quick look at each Chapter's contents before hanging our clothes on the hickory limb and diving into the shark infested waters. Maybe, just maybe depending upon your intellectual curiosity and persistence, the sharks will turn out to be pussy cats of the first water....no pun implied or intended. "A little knowledge is a dangerous thing, drink deep or taste not of the Pyrrhean spring." "Bull roar," the ancient philosopher said. Learning and understanding something new is better than learning nothing at all. Let's try it.

SUMMARY:

Chapter 1 starts off quite logically by presenting a system for disassembling Level II ROM using Level II BASIC's PEEK function. The entire Z-80 694 each opcodes and instruction set are covered using the TRS-80's decimal PEEK format. As such, those readers WITHOUT disassemblers may decode Level II ROM. Those readers with DISASSEMBLERs should nevertheless scan this Chapter to ensure that they fully understand the very broad capabilities of the Z-80 microprocessor.

Chapter 2 is a partially disassembled listing of Level II ROM with approximately 256 object codes and 256 source codes blanked out to protect the copyright owners. It is included as a necessary reference for those TRS-80 users who have already paid the licensing fee by purchasing Level II ROM. To others, it is totally useless.

Chapter 3 familiarizes the reader with the VARPTR function as used with integers including locating in MEM the variable's number NT (number type), identification, and decoding its value. A few assembly language ROM CALLs somewhat similar to the VARPTR function are discussed.

Chapter 4 addresses the VARPTR function when dealing with strings and string arrays of any depth. Strings are handled quite differently from integers and single/double precision numbers. String identification, location, and recall are covered in depth.

Chapter 5 investigates and explains a previous "black hole" in virtually every microcomputer book written to date by covering single and double precision/floating point normalized decimal handling, memory storage, and decoding from every aspect conceivable. When you finish this lengthy Chapter you may be able to lecture at most any recognized University as an EXPERT on the subject as many Professors of Computer Science totally misunderstand it. This is both a fun and MUST Chapter for all who wish to thoroughly understand their TRS-80's machinations when dealing with single and double precision numbers of any size, shape, or form.

Chapter 6 covers an equivalent 'JKL' LPRINT all of the video display on the line printer, program and fully explains how it operates on non-disk, non-NEWDOS+, TRS-80 systems. Those readers who have the 'JKL' LPRINT function already built-into their NEWDOS are nevertheless advised to study this Chapter as it covers a number of interesting assembly language salient features.

Chapter 7 is the "big daddy" of assembly language data movement programs for both video and non-video RAM memory. The highlights of this Chapter teach the user fundamental video memory management and semi-automatic RAM storage and recall techniques.

Chapter 8 is the "son-of big daddy" assembly language data movement program for video & RAM memory. In actuality, this fascinating Chapter allows the reader to create two completely separate video displays with separate inputs, scrolling, MEM storage if desired, recall, and shift from TRANSMIT to RECEIVE using the 'clear' key as the transmit/receive switch. This program is designed expressly for use with telephone MODEMs, Morse code, and/or radio teletype programs where the split-screen capability it offers is a decided plus...unless you wish two separate TRS-80s to do the same job.

Chapter 9 is the only Chapter that includes data repeated from Volume I. It is included in Volume II so that readers do not have to refer back and forth between two books for both CALL addresses and important ancillary CALL functions such as data movement, compares, data conversions, and math routines.

Chapter 10 is a unique Morse code TRANSMIT and RECEIVE program that includes virtually ALL Level II BASIC functions. Though very lengthy, it is included in Volume II as a review of BASIC for the advanced programmer....no more no less. Both Volume I and II of this handbook have presumed that you know and understand Level II BASIC absolutely COLD. Here is a useful and interesting program....of approximately 16,000 bytes.... divided into 2 parts, instructions and main program....with which you may check yourself. Approximately 1000 of these programs are in use throughout the world by radio amateurs with TRS-80s.

Chapter 11 is an another BASIC review program for you Television buffs to check your understanding of Level II BASIC on. Never use a preposition etc., except when it serves your purpose. This program does indeed serve an additional useful function in that it LPRINTS out the azimuth and elevation from your home's location to the major TV satellites in geosynchronous orbit in the earth's equatorial plane. All you need do is look up your home location's latitude and longitude on a road map and GO. If you understand the simple spherical trigonometric functions used, by all means skip this Chapter and take a GIANT step forward to Chapter 12 which will hopefully spoil your entire day by discerning exactly how much "sunk-in" from each Chapter.

Chapter 12 contains the self-test questions and is broken down into questions for each Chapter. We recommend you test yourself here after completing each Chapter, same as in Volume 1. After all, this is what self-programmed learning is all about. If you do not have the answer to the question right on the tip of your tongue, we strongly recommend you go back to the appropriate Chapter and re-read the subject matter at least one more time before curiosity drives you to looking up the answer in Chapter 13. If you do not choose to do so, the only one you will be CHEATING is yourself.....reflect upon that for a moment before you skip ahead and look up the answer?

Chapter 13 includes a VERY brief bibliography on texts generally available that may or may NOT be of use to the reader, PLUS the answers to the self-test questions on Chapters 1 through 11.

The publisher has asked us to include a disclaimer regarding the answers to the questions. Ok, we will do it. Our answers to the questions are not pluperfect, but are the best ones that a sometime microwave engineer can come up with on an off the cuff basis. Ask a color blind artist the color of the sky, and he will say, "gray." To him the sky is truly "gray."

If you see a different and better answer to our questions don't write and argue your point as we fully realize that from your perspective the sky may truly be "gray." If the author was 'right' all the time, he would not be writing TRS-80 Disassembled Handbooks.

You will note in the bibliography/book review section of this Chapter a much more benign approach to the task than Volume 1 of the "Disassembled Handbook For TRS-80." Why the sudden turn about? When Volume 1 was written, the author was a very "ANGRY YOUNG MAN." The cause for this anger was that NO ONE except Andrew Hildebrand had ever truly disassembled Level II ROM, 'AND ILLUSTRATED HOW TO USE IT.'

As such, the author wasted nearly 2 years of time with assembly language programming, writing it the ROTE way every other author (except Hildebrand) presented the subject. What a waste of time and effort.....and hence the resulting ANGER.

Now that the author has vented his pique in Volume 1, he will return to his typical middle-aged somnolent outlook and write about the 'good things' he sees in other authors and their books, rather than otherwise. "That goes for your cat too, and even Radio Shack."

Chapter 14, the appendix, includes a list of programs available on disk or cassette for both Volumes I and II, and an INDEX for Volumes I and II of the disassembled handbook. If you have forgotten EXACTLY where a useful bit or byte of information was located, this should be of assistance to you.

Robert M. Richardson May 1980

Chautauqua Lake, New York 14722

NOTE:

Both Volumes 1 and 2 of the "Disassembled Handbook For TRS-80" were written using a TRS-80, Shrayer's disk Electric Pencil (tm), the Western I/O IBM Selectric (tm) printer terminal, dual Wangco 82 mini-floppies by Siemens from Percom, and Mumford Micro's excellent 3 speed clock modification.

The above combination has proven vastly superior to the IBM 'Mag Card' and its predecessor magnetic tape system which the author used for nearly 10 years.

- CHAPTER 1 -

DISASSEMBLING OBJECT CODE - FUNDAMENTALS

INTRODUCTION:
A copyright is not a right given to the rich to rob the poor, nor vice versa. It is quite simply the result of 100's of years English common law translated into a law passed by our U.S. Congress that gives the copyright owner(s) the right to protect their investment in time, effort, and expense for an original work of sundry varieties, for a given period of time. During this period of time the copyright owner has the exclusive right to (hopefully) make a profit on the investment required to create the original work, whether it was a song, a book, a poem, or even a TRS-80 software program such as the Level II ROM BASIC written by Microsoft's Paul Allen and Bill Gates.

A copyright is not a license to steal nor a license to shroud in secrecy the basic laws of nature, including such things as the study and understanding of computer programming at the disassembled object code level. Now, we have what seems to be a paradox; i.e., "how do we study a copyrighted software program and at the same time protect the copyright owner(s) legal rights to the fruits of their work?" The meets and bounds of infringing a software program's copyright are ill defined at best in 1980 and the type of question that attorneys' love to ponder, on an hourly fee basis. The author has no intention whatsoever of violating Microsoft's Level II ROM copyright so here is the way we intend to protect Microsoft's copyright while at the same time writing Volume II of the "Disassembled Handbook for TRS-80."

We will print a disassembled version of Level II ROM that will NOT work unless you the reader have purchased the Level II ROM from Radio Shack and thereby implicitly paid the copyright owner, Microsoft, the copyright royalty due. We will accomplish this difficult feat by BLANKING OUT a considerable number of Level II ROM object and source code statements that are absolutely necessary for the program to work and let you, the purchaser of this copyrighted work, fill in the blank spaces using your own TRS-80's PEEK statements or disassembler if you have one. By your purchase of Level II ROM you most certainly have the legal right to do so for your own use, but you MAY NOT legally reproduce and sell this information, anymore than the right YOU DO NOT HAVE to xerox a copyrighted book you purchased and hawk it on the street corner. You may review the book, quote segments of the book, and even use the book in a study course IF your students do indeed purchase the book from the copyright owner. We ask your forbearance and patience for this difficult but necessary approach to protect Microsoft's rights. Now, let's get on with breaking the code.

The "Disassembled Handbook," both Volumes I and II presume that you, the reader, has a modest amount of assembly language programming experience. "Modest" is an ill defined word at best and means nothing more or less than the reader chooses to put into it. Cards, letters and phone calls from readers of Volume I varied from: "you assumed the reader was much more advanced than I am," to: "you spent way too much time on fundamentals." Surprisingly, the majority of feedback we received was: "what an enjoyable introduction to using Level II ROM functions. When will Volume II be published?"

Since we cannot possibly keep everyone happy all of the time, we'll continue the approach we used in Volume I. For you really advanced 'hot-doggers' out there in computerland, you have the option of skipping any Chapter you wish. Be our guest and by all means do so, though we suggest you use the self-test questions and answers to ascertain the correctness of your self esteem before blithely charging ahead. Many fighter pilots assured of their infallibility are enjoying endless peace and solitude at Arlington Cemetery while us more timid fighter pilots are suffering from old age and pecking away at TRS-80 keyboards while our grandchildren scurry around us.

Dr. Federico Faggin, creator of the 8008, 8080, and Z-80 microprocessors probably had no idea whatsoever in the late 1960's and early 1970's that he would be the founder, the Orville & Wilbur if you will, of what has turned out to be one of the most significant technological triumphs of this century. His reward? He is founder and President of Zilog, an EXXON affiliate. If you had the opportunity to communicate with either an 8008 or 8080 microprocessor you will realize what a GIANT step forward the Z-80 microprocessor was. The Z-80 is far more than a "son of an 8080." It is truly an entirely new ball game with an instruction set more than double its predecessor, the 8080. Some knowledgable engineers state that the power of a microprocessor varies as the square of its instruction set. If so, then the Z-80 is more than 4 times as powerful than an 8080. Let that sink in a moment. With all the new 16, 32, and even 64 bit word micros now coming down the pike, will not the Z-80 be just a passing fad? "No way," the old philospher said. "Mark my word, they'll still be producing Z-80's through the end of the century." Let's compare the Z-80 to the Boeing 727 tri-jet airliner STILL in production after nearly 20 years. Incredible, but true.

Form, fit, function.....the Z-80's got them all, just like the 727.

DECODING OBJECT CODES' INSTRUCTIONS WITH THE PEEK FUNCTION:

Z-80 object codes may consist of anywhere from 1 to 4 bytes in length. The Z-80 instruction set consists of 694 different object codes. The following pages in this Chapter include all 694 object codes and their instructions' source codes.

Just how does one ascertain the complete object code and its source code instruction if only the address in Level II ROM is known for the FIRST byte of the object code? Answer: quite easily using the following pages of "Z-80 Object Codes and Instructions." We will assume for the time being that the reader DOES NOT have a disassembler, and proceed accordingly using only the PEEK function of Level II BASIC.

The FIRST byte of the object code is our signpost in every case. Since we are using the PEEK function, all numbers shown on the video display will be in decimal. As such, all numbers will be given in decimal (and hex where there is room) in this particular discussion. Here is the logic we will use in deciphering the source code instruction from the object code:

1. The first byte of the object code delineates the specific instruction given in the next 3 pages, UNLESS that byte = 203, 221, 237, or 253.
2. If the first byte of the object code = 203, THEN GOTO the pages covering 203+ the 2nd byte of the object code.
3. If the first byte of the object code = 221, THEN GOTO the pages covering 221+ the 2nd & 4th byte of the object code.
4. If the first byte of the object code = 237, THEN GOTO the pages covering 237+ the 2nd byte of the object code.
5. If the first byte of the object code = 253, THEN GOTO the pages covering 253+ the 2nd & 4th byte of the object code.

The rules of this game are simple enough and will allow you to disassemble ANY object code in Level II ROM whose beginning address you know. What it will NOT disassemble CORRECTLY is the source code/instructions for constants, address table entry points, and data lists in ROM, but neither will ANY OTHER DISASSEMBLER. Not to fret though, as the blanked out Level II object codes and instructions in the next Chapter are all single byte honest to goodness REAL members of the Z-80 instruction set.

Actually, this form of manually using the PEEK function to decode Level II ROM is called the "4D system." The four D's stand for: dirty-dog difficult-drudgery. Admittedly it is not as easy as loading the "DISASSEM" from disk, but we have included it so that those readers WITHOUT disassemblers may decode those BLANKED OUT object codes and instructions in Chapter 2 using their own "copyright paid" Level II ROM PEEK function. Even if you do have a disassembler we suggest you read through the rest of this Chapter as you may learn something new.

OBJECT CODES/INSTRUCTIONS IN DECIMAL AND HEX NUMERICAL ORDER

DEC.	HEX.	INSTRUCTION		DEC.	HEX.	INSTRUCTION	
0	00	NOP		1	01	LD	BC,data
2	02	LD	(BC),A	3	03	INC	BC
4	04	INC	B	5	05	DEC	B
6	06	LD	B,data	7	07	RLCA	
8	08	EX	AF,AF'	9	09	ADD	HL,BC
10	0A	LD	A,(BC)	11	0B	DEC	BC
12	0C	INC	C	13	0D	DEC	C
14	0E	LD	C,data	15	0F	RRCA	
16	10	DJNZ	<256	17	11	LD	DE,data
18	12	LD	(DE),A	19	13	INC	DE
20	14	INC	D	21	15	DEC	D
22	16	LD	D,data	23	17	RLA	
24	18	JR	<256	25	19	ADD	HL,DE
26	1A	LD	A,(DE)	27	1B	DEC	DE
28	1C	INC	E	29	1D	DEC	E
30	1E	LD	E,data	31	1F	RRA	
32	20	JR	NZ,<256	33	21	LD	HL,data
34	22	LD	(addr),HL	35	23	INC	HL
36	24	INC	H	37	25	DEC	H
38	26	LD	H,data	39	27	DAA	
40	28	JR	Z,<256	41	29	ADD	HL,HL
42	2A	LD	HL,(addr)	43	2B	DEC	HL
44	2C	INC	L	45	2D	DEC	L
46	2E	LD	L,data	47	2F	CPL	
48	30	JR	NC,<256	49	31	LD	SP,data
50	32	LD	(addr),A	51	33	INC	SP
52	34	INC	(HL)	53	35	DEC	(HL)
54	36	LD	(HL),data	55	37	SCF	
56	38	JR	C,<256	57	39	ADD	HL,SP
58	3A	LD	A,(addr)	59	3B	DEC	SP
60	3C	INC	A	61	3D	DEC	A
62	3E	LD	A,data	63	3F	CCF	
64	40	LD	B,B	65	41	LD	B,C
66	42	LD	B,D	67	43	LD	B,E
68	44	LD	B,H	69	45	LD	B,L
70	46	LD	B,(HL)	71	47	LD	B,A
72	48	LD	C,B	73	49	LD	C,C
74	4A	LD	C,D	75	4B	LD	C,E
76	4C	LD	C,H	77	4D	LD	C,L
78	4E	LD	C,(HL)	79	4F	LD	C,A
80	50	LD	D,B	81	51	LD	D,C
82	52	LD	D,D	83	53	LD	D,E
84	54	LD	D,H	85	55	LD	D,L
86	56	LD	D,(HL)	87	57	LD	D,A
88	58	LD	E,B	89	59	LD	E,C
90	5A	LD	E,D	91	5B	LD	E,E
92	5C	LD	E,H	93	5D	LD	E,L
94	5E	LD	E,(HL)	95	5F	LD	E,A
96	60	LD	H,B	97	61	LD	H,C
98	62	LD	H,D	99	63	LD	H,E

- OBJECT CODES/INSTRUCTIONS - IN DECIMAL AND HEX NUMBERICAL ORDER

DEC.	HEX.	INSTRUCTION		DEC.	HEX.	INSTRUCTION	
100	64	LD	H,H	101	65	LD	H,L
102	66	LD	H,(HL)	103	67	LD	H,A
104	68	LD	L,B	105	69	LD	L,C
106	6A	LD	L,D	107	6B	LD	L,E
108	6C	LD	L,H	109	6D	LD	L,L
110	6E	LD	L,(HL)	111	6F	LD	L,A
112	70	LD	(HL),B	113	71	LD	(HL),C
114	72	LD	(HL),D	115	73	LD	(HL),E
116	74	LD	(HL),H	117	75	LD	(HL),L
118	76	HALT		119	77	LD	(HL),A
120	78	LD	A,B	121	79	LD	A,C
122	7A	LD	A,D	123	7B	LD	A,E
124	7C	LD	A,H	125	7D	LD	A,L
126	7E	LD	A,(HL)	127	7F	LD	A,A
128	80	ADD	A,B	129	81	ADD	A,C
130	82	ADD	A,D	131	83	ADD	A,E
132	84	ADD	A,H	133	85	ADD	A,L
134	86	ADD	A,(HL)	135	87	ADD	A,A
136	88	ADC	A,B	137	89	ADC	A,C
138	8A	ADC	A,D	139	8B	ADC	A,E
140	8C	ADC	A,H	141	8D	ADC	A,L
142	8E	ADC	A,(HL)	143	8F	ADC	A,A
144	90	SUB	B	145	91	SUB	C
146	92	SUB	D	147	93	SUB	E
148	94	SUB	H	149	95	SUB	L
150	96	SUB	(HL)	151	97	SUB	A
152	98	SBC	A,B	153	99	SBC	A,C
154	9A	SBC	A,D	155	9B	SBC	A,E
156	9C	SBC	A,H	157	9D	SBC	A,L
158	9E	SBC	A,(HL)	159	9F	SBC	A,A
160	A0	AND	B	161	A1	AND	C
162	A2	AND	D	163	A3	AND	E
164	A4	AND	H	165	A5	AND	L
166	A6	AND	(HL)	167	A7	AND	A
168	A8	XOR	B	169	A9	XOR	C
170	AA	XOR	D	171	AB	XOR	E
172	AC	XOR	H	173	AD	XOR	L
174	AE	XOR	(HL)	175	AF	XOR	A
176	B0	OR	B	177	B1	OR	C
178	B2	OR	D	179	B3	OR	E
180	B4	OR	H	181	B5	OR	L
182	B6	OR	(HL)	183	B7	OR	A
184	B8	CP	B	185	B9	CP	C
186	BA	CP	D	187	BB	CP	E
188	BC	CP	H	189	BD	CP	L
190	BE	CP	(HL)	191	BF	CP	A
192	C0	RET	NZ	193	C1	POP	BC
194	C2	JP	NZ,addr	195	C3	JP	addr
196	C4	CALL	NZ,addr	197	C5	PUSH	BC
198	C6	ADD	A,data	199	C7	RST	00H

- OBJECT CODES/INSTRUCTIONS - IN DECIMAL AND HEX NUMBERICAL ORDER

DEC.	HEX.	INSTRUCTION		DEC.	HEX.	INSTRUCTION	
200	C8	RET	Z	201	C9	RET	
202	CA	JP	Z,addr	203	CB	* * * * * *	
204	CC	CALL	Z,addr	205	CD	CALL	addr
206	CE	ADC	A,data	207	CF	RST	08H
208	D0	RET	NC	209	D1	POP	DE
210	D2	JP	NC,addr	211	D3	OUT	(PORT),A
212	D4	CALL	NC,addr	213	D5	PUSH	DE
214	D6	SUB	data	215	D7	RST	10H
216	D8	RET	C	217	D9	EXX	
218	DA	JP	C,addr	219	DB	IN	A,(PORT)
220	DC	CALL	C,addr	221	DD	* * * * * *	
222	DE	SBC	A,data	223	DF	RST	18H
224	E0	RET	PO	225	E1	POP	HL
226	E2	JP	PO,addr	227	E3	EX	(SP),HL
228	E4	CALL	PO,addr	229	E5	PUSH	HL
230	E6	AND	data	231	E7	RST	20H
232	E8	RET	PE	233	E9	JP	(HL)
234	EA	JP	PE,addr	235	EB	EX	DE,HL
236	EC	CALL	PE,addr	237	ED	* * * * * *	
238	EE	XOR	data	239	EF	RST	28H
240	F0	RET	P	241	F1	POP	AF
242	F2	JP	P,addr	243	F3	DI	
244	F4	CALL	P,addr	245	F5	PUSH	AF
246	F6	OR	data	247	F7	RST	30H
248	F8	RET	M	249	F9	LD	SP,HL
250	FA	JP	M,addr	251	FB	EI	
252	FC	CALL	M,addr	253	FD	* * * * * *	
254	FE	CP	data	255	FF	RST	38H

* * * * * *

NOTE:

- IF first byte = 203 THEN GOTO page 14.
- IF first byte = 221 THEN GOTO page 18.
- IF first byte = 237 THEN GOTO page 19.
- IF first byte = 253 THEN GOTO page 20.

- IF FIRST BYTE = 203 -

SECOND BYTE DEC.	HEX.	INSTRUCTION		SECOND BYTE DEC.	HEX.	INSTRUCTION	
0	00	RLC	B	1	01	RLC	C
2	02	RLC	D	3	03	RLC	E
4	04	RLC	H	5	05	RLC	L
6	06	RLC	(HL)	7	07	RLC	A
8	08	RRC	B	9	09	RRC	C
10	0A	RRC	D	11	0B	RRC	E
12	0C	RRC	H	13	0D	RRC	L
14	0E	RRC	(HL)	15	0F	RRC	A
16	10	RL	B	17	11	RL	C
18	12	RL	D	19	13	RL	E
20	14	RL	H	21	15	RL	L
22	16	RL	(HL)	23	17	RL	A
24	18	RR	B	25	19	RR	C
26	1A	RR	D	27	1B	RR	E
28	1C	RR	H	29	1D	RR	L
30	1E	RR	(HL)	31	1F	RR	A
32	20	SLA	B	33	21	SLA	C
34	22	SLA	D	35	23	SLA	E
36	24	SLA	H	37	25	SLA	L
38	26	SLA	(HL)	39	27	SLA	A
40	28	SRA	B	41	29	SRA	C
42	2A	SRA	D	43	2B	SRA	E
44	2C	SRA	H	45	2D	SRA	L
46	2E	SRA	(HL)	47	2F	SRA	A
56	38	SRL	B	57	39	SRL	C
58	3A	SRL	D	59	3B	SRL	E
60	3C	SRL	H	61	3D	SRL	L
62	3E	SRL	(HL)	63	3F	SRL	A
64	40	BIT	0,B	65	41	BIT	0,C
66	42	BIT	0,D	67	43	BIT	0,E
68	44	BIT	0,H	69	45	BIT	0,L
70	46	BIT	0,(HL)	71	47	BIT	0,A
72	48	BIT	1,B	73	49	BIT	1,C
74	4A	BIT	1,D	75	4B	BIT	1,E
76	4C	BIT	1,H	77	4D	BIT	1,L
78	4E	BIT	1,(HL)	79	4F	BIT	1,A
80	50	BIT	2,B	81	51	BIT	2,C
82	52	BIT	2,D	83	53	BIT	2,E
84	54	BIT	2,H	85	55	BIT	2,L
86	56	BIT	2,(HL)	87	57	BIT	2,A
88	58	BIT	3,B	89	59	BIT	3,C
90	5A	BIT	3,D	91	5B	BIT	3,E
92	5C	BIT	3,H	93	5D	BIT	3,L
94	5E	BIT	3,(HL)	95	5F	BIT	3,A
96	60	BIT	4,B	97	61	BIT	4,C
98	62	BIT	4,D	99	63	BIT	4,E

NOTE: DECIMAL 48 TO 55 MISSING IS CORRECT.

- IF FIRST BYTE = 203 -

SECOND BYTE DEC.	HEX.	INSTRUCTION		SECOND BYTE DEC.	HEX.	INSTRUCTION	
100	64	BIT	4,H	101	65	BIT	4,L
102	66	BIT	4,(HL)	103	67	BIT	4,A
104	68	BIT	5,B	105	69	BIT	5,C
106	6A	BIT	5,D	107	6B	BIT	5,E
108	6C	BIT	5,H	109	6D	BIT	5,L
110	6E	BIT	5,(HL)	111	6F	BIT	5,A
112	70	BIT	6,B	113	71	BIT	6,C
114	72	BIT	6,D	115	73	BIT	6,E
116	74	BIT	6,H	117	75	BIT	6,L
118	76	BIT	6,(HL)	119	77	BIT	6,A
120	78	BIT	7,B	121	79	BIT	7,C
122	7A	BIT	7,D	123	7B	BIT	7,E
124	7C	BIT	7,H	125	7D	BIT	7,L
126	7E	BIT	7,(HL)	127	7F	BIT	7,A
128	80	RES	0,B	129	81	RES	0,C
130	82	RES	0,D	131	83	RES	0,E
132	84	RES	0,H	133	85	RES	0,L
134	86	RES	0,(HL)	135	87	RES	0,A
136	88	RES	1,B	137	89	RES	1,C
138	8A	RES	1,D	139	8B	RES	1,E
140	8C	RES	1,H	141	8D	RES	1,L
142	8E	RES	1,(HL)	143	8F	RES	1,A
144	90	RES	2,B	145	91	RES	2,C
146	92	RES	2,D	147	93	RES	2,E
148	94	RES	2,H	149	95	RES	2,L
150	96	RES	2,(HL)	151	97	RES	2,A
152	98	RES	3,B	153	99	RES	2,C
154	9A	RES	3,D	155	9B	RES	3,E
156	9C	RES	3,H	157	9D	RES	3,L
158	9E	RES	3,(HL)	159	9F	RES	3,A
160	A0	RES	4,B	161	A1	RES	4,C
162	A2	RES	4,D	163	A3	RES	4,E
164	A4	RES	4,H	165	A5	RES	4,L
166	A6	RES	4,(HL)	167	A7	RES	4,A
168	A8	RES	5,B	169	A9	RES	5,C
170	AA	RES	5,D	171	AB	RES	5,E
172	AC	RES	5,H	173	AD	RES	5,L
174	AE	RES	5,(HL)	175	AF	RES	5,A
176	B0	RES	6,B	177	B1	RES	6,C
178	B2	RES	6,D	179	B3	RES	6,E
180	B4	RES	6,H	181	B5	RES	6,L
182	B6	RES	6,(HL)	183	B7	RES	6,A
184	B8	RES	7,B	185	B9	RES	7,C
186	BA	RES	7,D	187	BB	RES	7,E
188	BC	RES	7,H	189	BD	RES	7,L
190	BE	RES	7,(HL)	191	BF	RES	7,A
192	C0	SET	0,B	193	C1	SET	0,C
194	C2	SET	0,D	195	C3	SET	0,E
196	C4	SET	0,H	197	C5	SET	0,L
198	C6	SET	0,(HL)	199	C7	SET	0,A

- IF FIRST BYTE = 203 - (CB H)

SECOND BYTE DEC.	HEX.	INSTRUCTION		SECOND BYTE DEC.	HEX.	INSTRUCTION	
200	C8	SET	1,B	201	C9	SET	1,C
202	CA	SET	1,D	203	CB	SET	1,E
204	CC	SET	1,H	205	CD	SET	1,L
206	CE	SET	1,(HL)	207	CF	SET	1,A
208	D0	SET	2,B	209	D1	SET	2,C
210	D2	SET	2,D	211	D3	SET	2,E
212	D4	SET	2,H	213	D5	SET	2,L
214	D6	SET	2,(HL)	215	D7	SET	2,A
216	D8	SET	3,B	217	D9	SET	3,C
218	DA	SET	3,D	219	DB	SET	3,E
220	DC	SET	3,H	221	DD	SET	3,L
222	DE	SET	3,(HL)	223	DF	SET	3,A
224	E0	SET	4,B	225	E1	SET	4,C
226	E2	SET	4,D	227	E3	SET	4,E
228	E4	SET	4,H	229	E5	SET	4,L
230	E6	SET	4,(HL)	231	E7	SET	4,A
232	E8	SET	5,B	233	E9	SET	5,C
234	EA	SET	5,D	235	EB	SET	5,E
236	EC	SET	5,H	237	ED	SET	5,L
238	EE	SET	5,(HL)	239	EF	SET	5,A
240	F0	SET	6,B	241	F1	SET	6,C
242	F2	SET	6,D	243	F3	SET	6,E
244	F4	SET	6,H	245	F5	SET	6,L
246	F6	SET	6,(HL)	247	F7	SET	6,A
248	F8	SET	7,B	249	F9	SET	7,C
250	FA	SET	7,D	251	FB	SET	7,E
252	FC	SET	7,H	253	FD	SET	7,L
254	FE	SET	7,(HL)	255	FF	SET	7,A

READY
>
COMMENTS ON Z-80 OBJECT CODES & SOURCE STATEMENTS:

The symmetry of the Z-80 object codes and its assembler's source statements are a thing of beauty and a joy forever once one gets beyond the first 255 object codes. Why? Dr. Faggin and his team that developed the Z-80 insisted that the Z-80 be upward compatible with the old 8080 microprocessor. As such, they were 'stuck' with most of the first 255 object codes, but free to embellish all thereafter. Their logic, as illustrated above, is indeed a thing of grace and beauty.

Ok, you undoubtedly guessed it. The above object codes and their corresponding source code statements on this page were automatically typed out by our trustworthy Western I/O IBM Selectric printer terminal from a relatively simple BASIC software program. Here's an extra credit question for those of you who enjoy challenges. Write a software program in BASIC that will print the above data EXACTLY as shown, less column titles. No PEEKs, POKEs, sneaky VARPTRs, or machine language tricks are allowed. Just simple BASIC. The skills involved in writing BASIC are directly transferable to assembly language programming. Answer is on the next page.

- BASIC PROGRAM TO PRINT DATA ON PREVIOUS PAGE -

(CAN YOU CLEAN IT UP AND STREAMLINE IT? SURE YOU CAN.)

```
10 CLS:G=1:F=1
15 FORX=200TO255:PRINTX;"    ";:A=X
20 B=A/16:C=INT(B):D=A-16*C
25 IFC=0THENC$="0"ELSEIFC=1THENC$="1"ELSEIFC=2THENC$="2"ELSEIFC=
3THENC$="3"ELSEIFC=4THENC$="4"ELSEIFC=5THENC$="5"ELSEIFC=6THENC$
="6"ELSEIFC=7THENC$="7"
30 IFC=8THENC$="8"ELSEIFC=9THENC$="9"ELSEIFC=10THENC$="A"ELSEIFC
=11THENC$="B"ELSEIFC=12THENC$="C"ELSEIFC=13THENC$="D"ELSEIFC=14T
HENC$="E"ELSEIFC=15THENC$="F"
35 PRINTC$;
40 E=E+1:IFE=1THENC=D:GOTO25
45 IFE=2THENE=0:PRINT"    SET    ";
50 L=L+1:IFL=9THENG=G+1:L=0
55 IFF=1THENH$="B"ELSEIFF=2THENH$="C"ELSEIFF=3THENH$="D"ELSEIFF=
4THENH$="E"ELSEIFF=5THENH$="H"ELSEIFF=6THENH$="L"ELSEIFF=7THENH$
="(HL)"ELSEIFF=8THENH$="A"
60 F=F+1:IFF=9THENF=1
65 G$=STR$(G):PRINTG$;",";H$;
70 J=J+1:IFJ=2THENPRINT:J=0
75 IFJ=1THENPRINT,;"    ";
80 NEXTX
```

PROGRAM REVIEW:

What have we got here? Not much. Just a simple FOR - NEXT loop in lines 15 and 80. Lines 20 to 40 are a basic BASIC decimal to hexadecimal conversion subroutine for any decimal numbers from zero to 255 that are printed out in hex in line 35. The rest of this brief program are simple counters with line 55 an IF - THEN - ELSE subroutine that stuffs the appropriate registers and (HL) into the proper places.

Are there other ways and means of writing this program in BASIC? You bet there are....lots and lots. This is only one simplistic approach to the problem at hand.

How to grade yourself? Very easily. If the program you wrote prints out the EXACT SAME DATA with all the spaces and so forth in the EXACT SAME position on video, then you win.

If your program prints out on the video display extra spaces between ANY of the DATA, then you lose. Go back and try again until you thoroughly understand the whys and wherefores of what you're doing.

REMEMBER: A skillful BASIC programmer can invariably become a skillful assembly language programmer with only modest practice and effort. The message here is: DO NOT BLITHELY SKIP OVER THE BASICS.

- IF FIRST BYTE = 221 - (DD H)

2ND BYTE DEC.	4TH BYTE DEC.	INSTRUCTION		2ND BYTE DEC.	4TH BYTE DEC.	INSTRUCTION	
9	n/a	ADD	IX,BC	25	n/a	ADD	IX,DE
33	n/a	LD	IX,data	34	n/a	LD	(addr),IX
35	n/a	INC	IX	41	n/a	ADD	IX,IX
42	n/a	LD	IX,(addr)	43	n/a	DEC	IX
52	n/a	INC	(IX+zz)	53	n/a	DEC	(IX+zz),N
57	n/a	ADD	IX,SP	70	n/a	LD	B,(IX+zz)
78	n/a	LD	C,(IX+zz)	86	n/a	LD	D,(IX+zz)
94	n/a	LD	E,(IX+zz)	102	n/a	LD	H,(IX+zz)
110	n/a	LD	L,(IX+zz)	112	n/a	LD	(IX+zz),B
113	n/a	LD	(IX+zz),C	114	n/a	LD	(IX+zz),D
115	n/a	LD	(IX+zz),E	116	n/a	LD	(IX+zz),H
117	n/a	LD	(IX+zz),L	119	n/a	LD	(IX+zz),A
126	n/a	LD	A,(IX+zz)	134	n/a	ADD	A,(IX+zz)
142	n/a	ADC	A,(IX+zz)	150	n/a	SUB	(IX+zz)
158	n/a	SBC	A,(IX+zz)	166	n/a	AND	(IX+zz)
174	n/a	XOR	(IX+zz)	182	n/a	OR	(IX+zz)
190	n/a	CP	(IX+zz)	203	6	RLC	(IX+zz)
203	14	RRC	(IX+zz)	203	22	RL	(IX+zz)
203	30	RR	(IX+zz)	203	38	SLA	(IX+zz)
203	46	SRA	(IX+zz)	203	62	SRL	(IX+zz)
203	70	BIT	0,(IX+zz)	203	78	BIT	1,(IX+zz)
203	86	BIT	2,(IX+zz)	203	94	BIT	3,(IX+zz)
203	102	BIT	4,(IX+zz)	203	110	BIT	5,(IX+zz)
203	118	BIT	6,(IX+zz)	203	126	BIT	7,(IX+zz)
203	134	RES	0,(IX+zz)	203	142	RES	1,(IX+zz)
203	150	RES	2,(IX+zz)	203	158	RES	3,(IX+zz)
203	166	RES	4,(IX+zz)	203	174	RES	5,(IX+zz)
203	182	RES	6,(IX+zz)	203	190	RES	7,(IX+zz)
203	198	SET	0,(IX+zz)	203	206	SET	1,(IX+zz)
203	214	SET	2,(IX+zz)	203	222	SET	3,(IX+zz)
203	230	SET	4,(IX+zz)	203	238	SET	5,(IX+zz)
203	246	SET	6,(IX+zz)	203	254	SET	7,(IX+zz)
225	n/a	POP	IX	227	n/a	EX	(SP),IX
229	n/a	PUSH	IX	233	n/a	JP	(IX)
249	n/a	LD	SP,IX				
54	n/a	LD	(IX+zz),N				

NOTE:
zz = 8 bit signed binary address displacement.
N = 8 bit binary data unit.

- IF FIRST BYTE = 237 - (ED H)

SECOND BYTE		INSTRUCTION		SECOND BYTE		INSTRUCTION	
DEC.	HEX.			DEC.	HEX.		
64	40	IN	B,(C)	65	41	OUT	(C),B
66	42	SBC	HL,BC	67	43	LD	(addr),BC
68	44	NEG		69	45	RETN	
70	46	IM	0	71	47	LD	I,A
72	48	IN	C,(C)	73	49	OUT	(C),C
74	4A	ADC	HL,BC	75	4B	LD	BC,(addr)
77	4D	RETI		80	50	IN	D,(C)
81	51	OUT	(C),D	82	52	SBC	HL,DE
83	53	LD	(addr),DE	86	56	IM	1
87	57	LD	A,I	88	58	IN	E,(C)
89	59	OUT	(C),E	90	5A	ADC	HL,DE
91	5B	LD	DE,(addr)	94	5E	IM	2
96	60	IN	H,(C)	97	61	OUT	(C),H
98	62	SBC	HL,HL	103	67	RRD	
104	68	IN	L,(C)	105	69	OUT	(C),L
106	6A	ADC	HL,HL	111	6F	RLD	
114	72	SBC	HL,SP	115	73	LD	(addr),SP
120	78	IN	A,(C)	121	79	OUT	(C),A
122	7A	ADC	HL,SP	123	7B	LD	SP,(addr)
160	A0	LDI		161	A1	CPI	
162	A2	INI		163	A3	OUTI	
168	A8	LDD		169	A9	CPD	
170	AA	IND		171	AB	OUTD	
176	B0	LDIR		177	B1	CPIR	
178	B2	INIR		179	B3	OTIR	
184	B8	LDDR		185	B9	CPDR	
186	BA	INDR		187	BB	INDR	

- IF FIRST BYTE = 253 - (FD H)

2ND BYTE DEC.	4TH BYTE DEC.	INSTRUCTION		2ND BYTE DEC.	4TH BYTE DEC.	INSTRUCTION	
9	n/a	ADD	IY,BC	25	n/a	ADD	IY,DE
33	n/a	LD	IY,data	34	n/a	LD	(addr),IY
35	n/a	INC	IY	41	n/a	ADD	IY,IY
42	n/a	LD	IY,(addr)	43	n/a	DEC	IY
52	n/a	INC	(IY+zz)	53	n/a	DEC	(IY+zz),N
57	n/a	ADD	IY,SP	70	n/a	LD	B,(IY+zz)
78	n/a	LD	C,(IY+zz)	86	n/a	LD	D,(IY+zz)
94	n/a	LD	E,(IY+zz)	102	n/a	LD	H,(IY+zz)
110	n/a	LD	L,(IY+zz)	112	n/a	LD	(IY+zz),B
113	n/a	LD	(IY+zz),C	114	n/a	LD	(IY+zz),D
115	n/a	LD	(IY+zz),E	116	n/a	LD	(IY+zz),H
117	n/a	LD	(IY+zz),L	119	n/a	LD	(IY+zz),A
126	n/a	LD	A,(IY+zz)	134	n/a	ADD	A,(IY+zz)
142	n/a	ADC	A,(IY+zz)	150	n/a	SUB	(IY+zz)
158	n/a	SBC	A,(IY+zz)	166	n/a	AND	(IY+zz)
174	n/a	XOR	(IY+zz)	182	n/a	OR	(IY+zz)
190	n/a	CP	(IY+zz)	203	6	RLC	(IY+zz)
203	14	RRC	(IY+zz)	203	22	RL	(IY+zz)
203	30	RR	(IY+zz)	203	38	SLA	(IY+zz)
203	46	SRA	(IY+zz)	203	62	SRL	(IY+zz)
203	70	BIT	0,(IY+zz)	203	78	BIT	1,(IY+zz)
203	86	BIT	2,(IY+zz)	203	94	BIT	3,(IY+zz)
203	102	BIT	4,(IY+zz)	203	110	BIT	5,(IY+zz)
203	118	BIT	6,(IY+zz)	203	126	BIT	7,(IY+zz)
203	134	RES	0,(IY+zz)	203	142	RES	1,(IY+zz)
203	150	RES	2,(IY+zz)	203	158	RES	3,(IY+zz)
203	166	RES	4,(IY+zz)	203	174	RES	5,(IY+zz)
203	182	RES	6,(IY+zz)	203	190	RES	7,(IY+zz)
203	198	SET	0,(IY+zz)	203	206	SET	1,(IY+zz)
203	214	SET	2,(IY+zz)	203	222	SET	3,(IY+zz)
203	230	SET	4,(IY+zz)	203	238	SET	5,(IY+zz)
203	246	SET	6,(IY+zz)	203	254	SET	7,(IY+zz)
225	n/a	POP	IY	227	n/a	EX	(SP),IY
229	n/a	PUSH	IY	233	n/a	JP	(IY)
249	n/a	LD	SP,IY	54	n/a	LD	(IY+22),N

NOTE:
zz = 8 bit signed binary address displacement.
N = 8 bit binary data unit.

QUESTION:
Does this page look similar to PAGE 18?

ANSWER:
It certainly should. It is identical except for the first byte of the object code and the fact that it deals solely with the IY index register instead of the IX index register. How beautiful and entirely logical. Like the Parthenon: symmetry - form - fit - function.

- CHAPTER 2 -

PARTIAL DISASSEMBLED LEVEL II ROM

INTRODUCTION:

The next some 60 odd pages are a partial printout of Level II ROM disassembled. As mentioned earlier, this printout will NOT work in any form or fashion unless you have purchased Level II ROM thereby paying the royalty due on the copyrighted software program.

Chapter 1 covers ALL Z-80 object codes and their corresponding source code/instructions in decimal numerical order so that readers without a disassembler may use the PEEK function to determine what data the blanked out memory locations contain.

It is not an easy task to perform and quite time consuming to decode the missing data, but absolutely necessary if you wish to learn and fully understand how all the myriad Level II ROM BASIC and ancillary functions work that you may now use, often with a single CALL.

The questions and answers section for Chapter 1 literally 'beats to death' the technique and practice of decoding a given MEM location's object code and corresponding source code/instruction. By all means review Chapter 1 till you have the system down pat BEFORE filling in the blanked out object and source codes in the following pages using the PEEK function with your royalty paid Level II ROM BASIC.

We suggest you approach the task from the beginning and logically proceed from the next page on through the balance of the partial listing of the object/source codes until all the blanked out spaces are filled in. Take your time. Don't rush the job, and you will have a perfect disassembled listing when you are finished.

Most importantly, DO NOT try and guess the next object code just because you have decoded a series of similar codes. If you do so, you will most certainly be WRONG as they are mixed up intentionally to protect Microsoft's copyright.

You will note that each column of partially disassembled Level II ROM has two (2) blanked out lines in randomly different locations. This means that you will have to decode approximately 256 different MEM locations to have a complete disassembled listing. Be patient as it is well worth the effort to be able to see how this beautifully written program by Bill Gates and Paul Allen operates and flows. This Level II ROM BASIC program has not become the defacto world standard BASIC with over 300,000 installations worldwide for no reason whatsoever.....other than it is the BEST.

```
0000 F3        DI
0001 AF        XOR   A
0002 C37406    JP    0674H
0005 C30040    JP    4000H
0008 C30040    JP    4000H
000B E1        POP   HL
000C E9        JP    (HL)
000D C39F06    JP    069FH
0010 C30340    JP    4003H
0013 C5        PUSH  BC
0014 0601      LD    B,01H
0016 182E      JR    0046H
0018 C30640    JP    4006H
001B C5        PUSH  BC
001C 0602      LD    B,02H
001E 1826      JR    0046H
0020 C30940    JP    4009H
0023 C5        PUSH  BC
0024 0604      LD    B,04H
0026 181E      JR    0046H
0028 C30C40    JP    400CH
002B 111540    LD    DE,4015H
002E 18E3      JR    0013H
0030 C30F40    JP    400FH
0033 111D40    LD    DE,401DH
0036 18E3      JR    001BH
0038 C31240    JP    4012H
003B 112540    LD    DE,4025H
003E 18DB      JR    001BH
0040 C3D905    JP    05D9H
0043
0044 00        NOP
0045 00        NOP
0046 C3C203    JP    03C2H
0049 CD2B00    CALL  002BH
004C B7        OR    A
004D C0        RET   NZ
004E 18F9      JR    0049H
0050 0D        DEC   C
0051 0D        DEC   C
0052 1F        RRA
0053 1F        RRA
0054 01015B    LD    BC,5B01H
0057 1B        DEC   DE
0058 0A        LD    A,(BC)
0059 1A        LD    A,(DE)
005A 08        EX    AF,AF'
005B 1809      JR    0066H
005D 19        ADD   HL,DE
005E 2020      JR    NZ,0080H
0060 0B        DEC   BC
0061 78        LD    A,B
0062 B1        OR    C
0063 20FB      JR    NZ,0060H
0065
0066 310006    LD    SP,0600H
0069 3AEC37    LD    A,(37ECH)
006C 3C        INC   A
006D FE02      CP    02H
006F D20000    JP    NC,0000H
0072 C3CC06    JP    06CCH
0075 118040    LD    DE,4080H
0078 21F718    LD    HL,18F7H
007B 012700    LD    BC,0027H
007E EDB0      LDIR
0080 21E541    LD    HL,41E5H
0083 363A      LD    (HL),3AH
0085
0086 70        LD    (HL),B
0087 23        INC   HL
0088 362C      LD    (HL),2CH
008A 23        INC   HL
008B 22A740    LD    (40A7H),HL
008E 112D01    LD    DE,012DH
0091 061C      LD    B,1CH
0093 215241    LD    HL,4152H
0096 36C3      LD    (HL),0C3H
0098 23        INC   HL
0099 73        LD    (HL),E
009A 23        INC   HL
009B 72        LD    (HL),D
009C 23        INC   HL
009D 10F7      DJNZ  0096H
009F 0615      LD    B,15H
00A1 36C9      LD    (HL),0C9H
00A3 23        INC   HL
00A4 23        INC   HL
00A5 23        INC   HL
00A6 10F9      DJNZ  00A1H
00A8 21E842    LD    HL,42E8H
00AB 70        LD    (HL),B
00AC 31F841    LD    SP,41F8H
00AF CD8F1B    CALL  1B8FH
00B2 CDC901    CALL  01C9H
00B5 210501    LD    HL,0105H
00B8 CDA728    CALL  28A7H
00BB CDB31B    CALL  1BB3H
00BE 38F5      JR    C,00B5H
00C0 D7        RST   10H
00C1 B7        OR    A
00C2 2012      JR    NZ,00D6H
00C4 214C43    LD    HL,434CH
00C7
00C8 7C        LD    A,H
00C9 B5        OR    L
00CA 281B      JR    Z,00E7H
00CC 7E        LD    A,(HL)
00CD 47        LD    B,A
00CE 2F        CPL
00CF 77        LD    (HL),A
00D0 BE        CP    (HL)
00D1 70        LD    (HL),B
00D2 28F3      JR    Z,00C7H
00D4 1811      JR    00E7H
00D6 CD5A1E    CALL  1E5AH
00D9 B7        OR    A
00DA C29719    JP    NZ,1997H
00DD EB        EX    DE,HL
00DE 2B        DEC   HL
00DF 3E8F      LD    A,8FH
```

```
00E1 46       LD   B,(HL)
00E2 77       LD   (HL),A
00E3 BE       CP   (HL)
00E4 70       LD   (HL),B
00E5 20CE     JR   NZ,00B5H
00E7 2B       DEC  HL
00E8 111444   LD   DE,4414H
00EB DF       RST  18H
00EC DA7A19   JP   C,197AH
00EF 11CEFF   LD   DE,0FFCEH
00F2 22B140   LD   (40B1H),HL
00F5 19       ADD  HL,DE
00F6 22A040   LD   (40A0H),HL
00F9 CD4D1B   CALL 1B4DH
00FC 211101   LD   HL,0111H
00FF CDA728   CALL 28A7H
0102 C3191A   JP   1A19H
0105 4D       LD   C,L
0106 45       LD   B,L
0107 4D       LD   C,L
0108 4F       LD   C,A
0109 52       LD   D,D
010A 59       LD   E,C
010B 2053     JR   NZ,0160H
010D 49       LD   C,C
010E 5A       LD   E,D
010F 45       LD   B,L
0110
0111 52       LD   D,D
0112 41       LD   B,C
0113 44       LD   B,H
0114 49       LD   C,C
0115 4F       LD   C,A
0116 2053     JR   NZ,016BH
0118 48       LD   C,B
0119 41       LD   B,C
011A 43       LD   B,E
011B 4B       LD   C,E
011C 204C     JR   NZ,016AH
011E 45       LD   B,L
011F 56       LD   D,(HL)
0120 45       LD   B,L
0121 4C       LD   C,H
0122 2049     JR   NZ,016DH
0124 49       LD   C,C
0125 2042     JR   NZ,0169H
0127 41       LD   B,C
0128 53       LD   D,E
0129 49       LD   C,C
012A 43       LD   B,E
012B 0D       DEC  C
012C
012D 1E2C     LD   E,2CH
012F C3A219   JP   19A2H
0132 D7       RST  10H
0133 AF       XOR  A
0134 013E80   LD   BC,803EH
0137 013E01   LD   BC,013EH
013A F5       PUSH AF
013B CF       RST  08H
013C 28CD     JR   Z,010BH
013E 1C       INC  E
013F 2B       DEC  HL
0140 FE80     CP   80H
0142 D24A1E   JP   NC,1E4AH
0145 F5       PUSH AF
0146 CF       RST  08H
0147 2C       INC  L
0148 CD1C2B   CALL 2B1CH
014B FE30     CP   30H
014D D24A1E   JP   NC,1E4AH
0150 16FF     LD   D,0FFH
0152 14       INC  D
0153 D603     SUB  03H
0155 30FB     JR   NC,0152H
0157 C603     ADD  03H
0159 4F       LD   C,A
015A F1       POP  AF
015B 87       ADD  A,A
015C 5F       LD   E,A
015D 0602     LD   B,02H
015F 7A       LD   A,D
0160 1F       RRA
0161 57       LD   D,A
0162 7B       LD   A,E
0163
0164 5F       LD   E,A
0165 10F8     DJNZ 015FH
0167 79       LD   A,C
0168 8F       ADC  A,A
0169 3C       INC  A
016A 47       LD   B,A
016B AF       XOR  A
016C 37       SCF
016D 8F       ADC  A,A
016E 10FD     DJNZ 016DH
0170 4F       LD   C,A
0171 7A       LD   A,D
0172 F63C     OR   3CH
0174 57       LD   D,A
0175 1A       LD   A,(DE)
0176 B7       OR   A
0177 FA7C01   JP   M,017CH
017A 3E80     LD   A,80H
017C 47       LD   B,A
017D F1       POP  AF
017E B7       OR   A
017F 78       LD   A,B
0180 2810     JR   Z,0192H
0182 12       LD   (DE),A
0183 FA8F01   JP   M,018FH
0186 79       LD   A,C
0187 2F       CPL
0188 4F       LD   C,A
0189 1A       LD   A,(DE)
018A A1       AND  C
018B 12       LD   (DE),A
018C CF       RST  08H
018D 29       ADD  HL,HL
018E
```

```
018F B1        OR   C
0190 18F9      JR   018BH
0192 A1        AND  C
0193 C6FF      ADD  0FFH
0195 9F        SBC  A,A
0196 E5        PUSH HL
0197 CD8D09    CALL 098DH
019A E1        POP  HL
019B 18EF      JR   018CH
019D D7        RST  10H
019E ES
019F 3A9940    LD   A,(4099H)
01A2 B7        OR   A
01A3 2006      JR   NZ,01ABH
01A5 CD5803    CALL 0358H
01A8 B7        OR   A
01A9 2811      JR   Z,01BCH
01AB F5        PUSH AF
01AC AF        XOR  A
01AD 329940    LD   (4099H),A
01B0 3C        INC  A
01B1 CD5728    CALL 2857H
01B4 F1        POP  AF
01B5 2AD440    LD   HL,(40D4H)
01B8 77        LD   (HL),A
01B9 C38428    JP   2884H
01BC 212819    LD   HL,1928H
01BF 222141    LD   (4121H),HL
01C2 3E03      LD   A,03H
01C4 32AF40    LD   (40AFH),A
01C7 E1        POP  HL
01C8 C9        RET
01C9 3E1C      LD   A,1CH
01CB CD3A03    CALL 033AH
01CE 3E1F      LD   A,1FH
01D0 C33A03    JP   033AH
01D3 ED5F      LD   A,R
01D5 32AB40    LD   (40ABH),A
01D8 C9        RET
01D9 2101FC    LD   HL,0FC01H
01DC CD2102    CALL 0221H
01DF 060B      LD   B,0BH
01E1 10FE      DJNZ 01E1H
01E3 2102FC    LD   HL,0FC02H
01E6 CD2102    CALL 0221H
01E9 060B      LD   B,0BH
01EB 10FE      DJNZ 01EBH
01ED 2100FC    LD   HL,0FC00H
01F0 CD2102    CALL 0221H
01F3 065C      LD   B,5CH
01F5 10FE      DJNZ 01F5H
01F7 C9        RET
01F8
01F9 2100FB    LD   HL,0FB00H
01FC 181B      JR   0219H
01FE 7E        LD   A,(HL)
01FF D623      SUB  23H
0201 3E00      LD   A,00H
0203 200D      JR   NZ,0212H
0205 CD012B    CALL 2B01H
0208 CF        RST  08H
0209 2C        INC  L
020A 7B        LD   A,E
020B A2        AND  D
020C C602      ADD  02H
020E D24A1E    JP   NC,1E4AH
0211 3D        DEC  A
0212 32E437    LD   (37E4H),A
0215 E5        PUSH HL
0216 2104FF    LD   HL,0FF04H
0219 CD2102    CALL 0221H
021C E1        POP  HL
021D
021E 2100FF    LD   HL,0FF00H
0221 3A3D40    LD   A,(403DH)
0224 A4        AND  H
0225 B5        OR   L
0226 D3FF      OUT  (0FFH),A
0228 323D40    LD   (403DH),A
022B C9        RET
022C 3A3F3C    LD   A,(3C3FH)
022F EE0A      XOR  0AH
0231 323F3C    LD   (3C3FH),A
0234 C9        RET
0235 C5        PUSH BC
0236 E5        PUSH HL
0237 0608      LD   B,08H
0239 CD4102    CALL 0241H
023C 10FB      DJNZ 0239H
023E E1        POP  HL
023F C1        POP  BC
0240
0241 C5        PUSH BC
0242 F5        PUSH AF
0243 DBFF      IN   A,(0FFH)
0245 17        RLA
0246 30FB      JR   NC,0243H
0248 0641      LD   B,41H
024A 10FE      DJNZ 024AH
024C CD1E02    CALL 021EH
024F 0676      LD   B,76H
0251 10FE      DJNZ 0251H
0253 DBFF      IN   A,(0FFH)
0255 47        LD   B,A
0256 F1        POP  AF
0257 CB10      RL   B
0259 17        RLA
025A F5        PUSH AF
025B CD1E02    CALL 021EH
025E F1        POP  AF
025F C1        POP  BC
0260 C9        RET
0261 CD6402    CALL 0264H
0264 E5        PUSH HL
0265 C5        PUSH BC
0266 D5        PUSH DE
0267 F5        PUSH AF
0268 0E08      LD   C,08H
026A 57        LD   D,A
026B CDD901    CALL 01D9H
```

```
026E 7A       LD   A,D
026F 07       RLCA
0270 57       LD   D,A
0271 300B     JR   NC,027EH
0273 CDD901   CALL 01D9H
0276 0D       DEC  C
0277 20F2     JR   NZ,026BH
0279
027A D1       POP  DE
027B C1       POP  BC
027C E1       POP  HL
027D C9       RET
027E 0687     LD   B,87H
0280 10FE     DJNZ 0280H
0282 18F2     JR   0276H
0284 CDFE01   CALL 01FEH
0287 06FF     LD   B,0FFH
0289 AF       XOR  A
028A CD6402   CALL 0264H
028D 10FB     DJNZ 028AH
028F 3EA5     LD   A,0A5H
0291 18D1     JR   0264H
0293 CDFE01   CALL 01FEH
0296
0297 AF       XOR  A
0298 CD4102   CALL 0241H
029B FEA5     CP   0A5H
029D 20F9     JR   NZ,0298H
029F 3E2A     LD   A,2AH
02A1 323E3C   LD   (3C3EH),A
02A4 323F3C   LD   (3C3FH),A
02A7 E1       POP  HL
02A8 C9       RET
02A9 CD1403   CALL 0314H
02AC 22DF40   LD   (40DFH),HL
02AF CDF801   CALL 01F8H
02B2 CDE241   CALL 41E2H
02B5 318842   LD   SP,4288H
02B8 CDFE20   CALL 20FEH
02BB 3E2A     LD   A,2AH
02BD CD2A03   CALL 032AH
02C0 CDB31B   CALL 1BB3H
02C3 DACC06   JP   C,06CCH
02C6 D7       RST  10H
02C7 CA9719   JP   Z,1997H
02CA FE2F     CP   2FH
02CC 284F     JR   Z,031DH
02CE CD9302   CALL 0293H
02D1 CD3502   CALL 0235H
02D4 FE55     CP   55H
02D6 20F9     JR   NZ,02D1H
02D8 0606     LD   B,06H
02DA 7E       LD   A,(HL)
02DB B7       OR   A
02DC 2809     JR   Z,02E7H
02DE CD3502   CALL 0235H
02E1 BE       CP   (HL)
02E2 20ED     JR   NZ,02D1H
02E4 23       INC  HL
02E5 10F3     DJNZ 02DAH
02E7 CD2C02   CALL 022CH
02EA CD3502   CALL 0235H
02ED FE78     CP   78H
02EF 28B8     JR   Z,02A9H
02F1 FE3C     CP   3CH
02F3 20F5     JR   NZ,02EAH
02F5 CD3502   CALL 0235H
02F8 47       LD   B,A
02F9 CD1403   CALL 0314H
02FC 85       ADD  A,L
02FD 4F       LD   C,A
02FE CD3502   CALL 0235H
0301 77       LD   (HL),A
0302 23       INC  HL
0303 81       ADD  A,C
0304 4F       LD   C,A
0305 10F7     DJNZ 02FEH
0307 CD3502   CALL 0235H
030A B9       CP   C
030B 28DA     JR   Z,02E7H
030D 3E43     LD   A,43H
030F 323E3C   LD   (3C3EH),A
0312 18D6     JR   02EAH
0314 CD3502   CALL 0235H
0317 6F       LD   L,A
0318 CD3502   CALL 0235H
031B 67       LD   H,A
031C
031D EB       EX   DE,HL
031E 2ADF40   LD   HL,(40DFH)
0321 EB       EX   DE,HL
0322 D7       RST  10H
0323 C45A1E   CALL NZ,1E5AH
0326 208A     JR   NZ,02B2H
0328 EB       EX   DE,HL
0329 E9       JP   (HL)
032A C5       PUSH BC
032B 4F       LD   C,A
032C CDC141   CALL 41C1H
032F 3A9C40   LD   A,(409CH)
0332 B7       OR   A
0333 79       LD   A,C
0334 C1       POP  BC
0335 FA6402   JP   M,0264H
0338 2062     JR   NZ,039CH
033A D5       PUSH DE
033B CD3300   CALL 0033H
033E F5       PUSH AF
033F CD4803   CALL 0348H
0342 32A640   LD   (40A6H),A
0345 F1       POP  AF
0346 D1       POP  DE
0347
0348 3A3D40   LD   A,(403DH)
034B E608     AND  08H
034D 3A2040   LD   A,(4020H)
0350 2803     JR   Z,0355H
0352 0F       RRCA
0353 E61F     AND  1FH
0355 E63F     AND  3FH
```

```
0357
0358 CDC441     CALL 41C4H
035B D5         PUSH DE
035C CD2B00     CALL 002BH
035F D1         POP  DE
0360 C9         RET
0361 AF         XOR  A
0362 329940     LD   (4099H),A
0365 32A640     LD   (40A6H),A
0368 CDAF41     CALL 41AFH
036B C5         PUSH BC
036C 2AA740     LD   HL,(40A7H)
036F 06F0       LD   B,0F0H
0371 CDD905     CALL 05D9H
0374 F5         PUSH AF
0375 48         LD   C,B
0376 0600       LD   B,00H
0378 09         ADD  HL,BC
0379 3600       LD   (HL),00H
037B 2AA740     LD   HL,(40A7H)
037E F1         POP  AF
037F C1         POP  BC
0380 2B         DEC  HL
0381 D8         RET  C
0382 AF         XOR  A
0383 C9         RET
0384 CD5803     CALL 0358H
0387 B7         OR   A
0388 C0         RET  NZ
0389 18F9       JR   0384H
038B AF         XOR  A
038C 329C40     LD   (409CH),A
038F 3A9B40     LD   A,(409BH)
0392 B7         OR   A
0393 C8         RET  Z
0394 3E0D       LD   A,0DH
0396 D5         PUSH DE
0397 CD9C03     CALL 039CH
039A D1         POP  DE
039B
039C F5         PUSH AF
039D D5         PUSH DE
039E C5         PUSH BC
039F 4F         LD   C,A
03A0 1E00       LD   E,00H
03A2 FE0C       CP   0CH
03A4 2810       JR   Z,03B6H
03A6 FE0A       CP   0AH
03A8 2003       JR   NZ,03ADH
03AA 3E0D       LD   A,0DH
03AC 4F         LD   C,A
03AD FE0D       CP   0DH
03AF 2805       JR   Z,03B6H
03B1 3A9B40     LD   A,(409BH)
03B4 3C         INC  A
03B5 5F         LD   E,A
03B6 7B         LD   A,E
03B7 329B40     LD   (409BH),A
03BA 79         LD   A,C
03BB CD3B00     CALL 003BH
03BE C1         POP  BC
03BF D1         POP  DE
03C0 F1         POP  AF
03C1 C9         RET
03C2 E5         PUSH H,L
03C3 DDE5       PUSH IX
03C5 D5         PUSH DE
03C6 DDE1       POP  IX
03C8 D5         PUSH DE
03C9 21DD03     LD   HL,03DDH
03CC E5         PUSH HL
03CD 4F         LD   C,A
03CE 1A         LD   A,(DE)
03CF A0         AND  B
03D0 B8         CP   B
03D1 C23340     JP   NZ,4033H
03D4 FE02       CP   02H
03D6 DD6E01     LD   L,(IX+01H)
03D9 DD6602     LD   H,(IX+02H)
03DC E9         JP   (HL)
03DD D1         POP  DE
03DE DDE1       POP  IX
03E0 E1         POP  HL
03E1 C1         POP  BC
03E2 C9         RET
03E3 213640     LD   HL,4036H
03E6 010138     LD   BC,3801H
03E9 1600       LD   D,00H
03EB 0A         LD   A,(BC)
03EC 5F         LD   E,A
03ED AE         XOR  (HL)
03EE 73         LD   (HL),E
03EF A3         AND  E
03F0 2008       JR   NZ,03FAH
03F2 14         INC  D
03F3 2C         INC  L
03F4 CB01       RLC  C
03F6 F2EB03     JP   P,03EBH
03F9 C9         RET
03FA 5F         LD   E,A
03FB 7A         LD   A,D
03FC 07         RLCA
03FD 07         RLCA
03FE 07         RLCA
03FF 57         LD   D,A
0400 0E01       LD   C,01H
0402 79         LD   A,C
0403 A3         AND  E
0404 2005       JR   NZ,040BH
0406 14         INC  D
0407 CB01       RLC  C
0409 18F7       JR   0402H
040B 3A8038     LD   A,(3880H)
040E 47         LD   B,A
040F 7A         LD   A,D
0410 C640       ADD  40H
0412 FE60       CP   60H
0414 3013       JR   NC,0429H
0416 CB08       RRC  B
0418 3031       JR   NC,044BH
```

```
041A C620      ADD  20H
041C 57        LD   D,A
041D 3A4038    LD   A,(3840H)
0420 E610      AND  10H
0422 2828      JR   Z,044CH
0424 7A        LD   A,D
0425 D660      SUB  60H
0427 1822      JR   044BH
0429 D670      SUB  70H
042B 3010      JR   NC,043DH
042D C640      ADD  40H
042F FE3C      CP   3CH
0431 3802      JR   C,0435H
0433 EE10      XOR  10H
0435 CB08      RRC  B
0437 3012      JR   NC,044BH
0439 EE10      XOR  10H
043B 180E      JR   044BH
043D
043E CB08      RRC  B
0440 3001      JR   NC,0443H
0442 3C        INC  A
0443 215000    LD   HL,0050H
0446 4F        LD   C,A
0447 0600      LD   B,00H
0449 09        ADD  HL,BC
044A 7E        LD   A,(HL)
044B 57        LD   D,A
044C 01AC0D    LD   BC,0DACH
044F CD6000    CALL 0060H
0452 7A        LD   A,D
0453 FE01      CP   01H
0455 C0        RET  NZ
0456 EF        RST  28H
0457 C9        RET
0458 DD6E03    LD   L,(IX+03H)
045B DD6604    LD   H,(IX+04H)
045E 383A      JR   C,049AH
0460 DD7E05    LD   A,(IX+05H)
0463
0464 2801      JR   Z,0467H
0466 77        LD   (HL),A
0467 79        LD   A,C
0468 FE20      CP   20H
046A DA0605    JP   C,0506H
046D FE80      CP   80H
046F 3035      JR   NC,04A6H
0471 FE40      CP   40H
0473 3808      JR   C,047DH
0475 D640      SUB  40H
0477 FE20      CP   20H
0479 3802      JR   C,047DH
047B D620      SUB  20H
047D CD4105    CALL 0541H
0480 7C        LD   A,H
0481 E603      AND  03H
0483 F63C      OR   3CH
0485 67        LD   H,A
0486 56        LD   D,(HL)
0487 DD7E05    LD   A,(IX+05H)
048A B7        OR   A
048B 2805      JR   Z,0492H
048D DD7205    LD   (IX+05H),D
0490 365F      LD   (HL),5FH
0492 DD7503    LD   (IX+03H),L
0495 DD7404    LD   (IX+04H),H
0498 79        LD   A,C
0499
049A DD7E05    LD   A,(IX+05H)
049D B7        OR   A
049E C0        RET  NZ
049F 7E        LD   A,(HL)
04A0 C9        RET
04A1 7D        LD   A,L
04A2 E6C0      AND  0C0H
04A4 6F        LD   L,A
04A5 C9        RET
04A6 FEC0      CP   0C0H
04A8 38D3      JR   C,047DH
04AA D6C0      SUB  0C0H
04AC 28D2      JR   Z,0480H
04AE 47        LD   B,A
04AF 3E20      LD   A,20H
04B1 CD4105    CALL 0541H
04B4 10F9      DJNZ 04AFH
04B6 18C8      JR   0480H
04B8 7E        LD   A,(HL)
04B9 DD7705    LD   (IX+05H),A
04BC C9        RET
04BD AF        XOR  A
04BE 18F9      JR   04B9H
04C0 21003C    LD   HL,3C00H
04C3 3A3D40    LD   A,(403DH)
04C6 E6F7      AND  0F7H
04C8 323D40    LD   (403DH),A
04CB D3FF      OUT  (0FFH),A
04CD C9        RET
04CE 2B        DEC  HL
04CF 3A3D40    LD   A,(403DH)
04D2 E608      AND  08H
04D4 2801      JR   Z,04D7H
04D6 2B        DEC  HL
04D7 3620      LD   (HL),20H
04D9 C9        RET
04DA 3A3D40    LD   A,(403DH)
04DD E608      AND  08H
04DF C4E204    CALL NZ,04E2H
04E2 7D        LD   A,L
04E3 E63F      AND  3FH
04E5 2B        DEC  HL
04E6 C0        RET  NZ
04E7 114000    LD   DE,0040H
04EA 19        ADD  HL,DE
04EB
04EC 23        INC  HL
04ED 7D        LD   A,L
04EE E63F      AND  3FH
04F0 C0        RET  NZ
04F1 11C0FF    LD   DE,0FFC0H
04F4 19        ADD  HL,DE
```

```
04F5
04F6 3A3D40    LD   A,(403DH)
04F9 F608      OR   08H
04FB 323D40    LD   (403DH),A
04FE D3FF      OUT  (0FFH),A
0500 23        INC  HL
0501 7D        LD   A,L
0502 E6FE      AND  0FEH
0504 6F        LD   L,A
0505 C9        RET
0506 118004    LD   DE,0480H
0509 D5        PUSH DE
050A FE08      CP   08H
050C 28C0      JR   Z,04CEH
050E FE0A      CP   0AH
0510 D8        RET  C
0511 FE0E      CP   0EH
0513 384F      JR   C,0564H
0515 28A1      JR   Z,04B8H
0517 FE0F      CP   0FH
0519 28A2      JR   Z,04BDH
051B FE17      CP   17H
051D 28D7      JR   Z,04F6H
051F FE18      CP   18H
0521 28B7      JR   Z,04DAH
0523 FE19      CP   19H
0525 28C5      JR   Z,04ECH
0527 FE1A      CP   1AH
0529 28BC      JR   Z,04E7H
052B FE1B      CP   1BH
052D 28C2      JR   Z,04F1H
052F FE1C      CP   1CH
0531 288D      JR   Z,04C0H
0533 FE1D      CP   1DH
0535 CAA104    JP   Z,04A1H
0538 FE1E      CP   1EH
053A 2837      JR   Z,0573H
053C FE1F      CP   1FH
053E 283C      JR   Z,057CH
0540
0541 77        LD   (HL),A
0542 23        INC  HL
0543 3A3D40    LD   A,(403DH)
0546 E608      AND  08H
0548 2801      JR   Z,054BH
054A 23        INC  HL
054B 7C        LD   A,H
054C FE40      CP   40H
054E C0        RET  NZ
054F 11C0FF    LD   DE,0FFC0H
0552 19        ADD  HL,DE
0553 E5        PUSH HL
0554 11003C    LD   DE,3C00H
0557 21403C    LD   HL,3C40H
055A C5        PUSH BC
055B 01C003    LD   BC,03C0H
055E EDB0      LDIR
0560 C1        POP  BC
0561 EB        EX   DE,HL
0562 1819      JR   057DH
0564 7D        LD   A,L
0565 E6C0      AND  0C0H
0567 6F        LD   L,A
0568 E5        PUSH HL
0569 114000    LD   DE,0040H
056C 19        ADD  HL,DE
056D 7C        LD   A,H
056E FE40      CP   40H
0570 28E2      JR   Z,0554H
0572 D1        POP  DE
0573
0574 54        LD   D,H
0575 7D        LD   A,L
0576 F63F      OR   3FH
0578 5F        LD   E,A
0579 13        INC  DE
057A 1804      JR   0580H
057C E5        PUSH HL
057D 110040    LD   DE,4000H
0580 3620      LD   (HL),20H
0582 23        INC  HL
0583 7C        LD   A,H
0584 BA        CP   D
0585 20F9      JR   NZ,0580H
0587 7D        LD   A,L
0588 BB        CP   E
0589 20F5      JR   NZ,0580H
058B E1        POP  HL
058C C9        RET
058D 79        LD   A,C
058E B7        OR   A
058F 2840      JR   Z,05D1H
0591 FE0B      CP   0BH
0593 280A      JR   Z,059FH
0595 FE0C      CP   0CH
0597 201B      JR   NZ,05B4H
0599 AF        XOR  A
059A DDB603    OR   (IX+03H)
059D 2815      JR   Z,05B4H
059F DD7E03    LD   A,(IX+03H)
05A2 DD9604    SUB  (IX+04H)
05A5 47        LD   B,A
05A6 CDD105    CALL 05D1H
05A9 20FB      JR   NZ,05A6H
05AB 3E0A      LD   A,0AH
05AD 32E837    LD   (37E8H),A
05B0 10F4      DJNZ 05A6H
05B2 1818      JR   05CCH
05B4 F5        PUSH AF
05B5 CDD105    CALL 05D1H
05B8 20FB      JR   NZ,05B5H
05BA F1        POP  AF
05BB 32E837    LD   (37E8H),A
05BE FE0D      CP   0DH
05C0 C0        RET  NZ
05C1 DD3404    INC  (IX+04H)
05C4 DD7E04    LD   A,(IX+04H)
05C7 DDBE03    CP   (IX+03H)
05CA 79        LD   A,C
05CB C0        RET  NZ
```

```
05CC DD360400   LD   (IX+04H),00H
05D0 C9         RET
05D1 3AE837     LD   A,(37E8H)
05D4 E6F0       AND  0F0H
05D6 FE30       CP   30H
05D8 C9         RET
05D9 E5         PUSH HL
05DA 3E0E       LD   A,0EH
05DC CD3300     CALL 0033H
05DF 48         LD   C,B
05E0 CD4900     CALL 0049H
05E3 FE20       CP   20H
05E5 3025       JR   NC,060CH
05E7 FE0D       CP   0DH
05E9 CA6206     JP   Z,0662H
05EC FE1F       CP   1FH
05EE 2829       JR   Z,0619H
05F0 FE01       CP   01H
05F2 286D       JR   Z,0661H
05F4 11E005     LD   DE,05E0H
05F7 D5         PUSH DE
05F8 FE08       CP   08H
05FA 2834       JR   Z,0630H
05FC FE18       CP   18H
05FE 282B       JR   Z,062BH
0600 FE09       CP   09H
0602 2842       JR   Z,0646H
0604 FE19       CP   19H
0606 2839       JR   Z,0641H
0608 FE0A       CP   0AH
060A C0
060B D1         POP  DE
060C 77         LD   (HL),A
060D 78         LD   A,B
060E B7         OR   A
060F 28CF       JR   Z,05E0H
0611 7E         LD   A,(HL)
0612 23         INC  HL
0613 CD3300     CALL 0033H
0616 05         DEC  B
0617 18C7       JR   05E0H
0619 CDC901     CALL 01C9H
061C 41         LD   B,C
061D E1         POP  HL
061E E5         PUSH HL
061F C3E005     JP   05E0H
0622 CD3006     CALL 0630H
0625 2B         DEC  HL
0626 7E         LD   A,(HL)
0627 23         INC  HL
0628 FE0A       CP   0AH
062A C8         RET  Z
062B 78         LD   A,B
062C B9         CP   C
062D 20F3       JR   NZ,0622H
062F C9
0630 78         LD   A,B
0631 B9         CP   C
0632 C8         RET  Z
0633 2B         DEC  HL
0634 7E         LD   A,(HL)
0635 FE0A       CP   0AH
0637 23         INC  HL
0638 C8         RET  Z
0639 2B         DEC  HL
063A 3E08       LD   A,08H
063C CD3300     CALL 0033H
063F 04         INC  B
0640 C9         RET
0641 3E17       LD   A,17H
0643 C33300     JP   0033H
0646 CD4803     CALL 0348H
0649 E607       AND  07H
064B 2F         CPL
064C 3C         INC  A
064D C608       ADD  08H
064F 5F         LD   E,A
0650 78         LD   A,B
0651 B7         OR   A
0652 C8         RET  Z
0653 3E20       LD   A,20H
0655 77         LD   (HL),A
0656 23
0657 D5         PUSH DE
0658 CD3300     CALL 0033H
065B D1         POP  DE
065C 05         DEC  B
065D 1D         DEC  E
065E C8         RET  Z
065F 18EF       JR   0650H
0661 37         SCF
0662 F5         PUSH AF
0663 3E0D       LD   A,0DH
0665 77         LD   (HL),A
0666 CD3300     CALL 0033H
0669 3E0F       LD   A,0FH
066B CD3300     CALL 0033H
066E 79         LD   A,C
066F 90         SUB  B
0670 47         LD   B,A
0671 F1         POP  AF
0672 E1         POP  HL
0673 C9         RET
0674 D3FF       OUT  (0FFH),A
0676 21D206     LD   HL,06D2H
0679 110040     LD   DE,4000H
067C 013600     LD   BC,0036H
067F EDB0       LDIR
0681 3D         DEC  A
0682 3D         DEC  A
0683 20F1       JR   NZ,0676H
0685 0627       LD   B,27H
0687 12         LD   (DE),A
0688 13
0689 10FC       DJNZ 0687H
068B 3A4038     LD   A,(3840H)
068E E604       AND  04H
0690 C27500     JP   NZ,0075H
0693 317D40     LD   SP,407DH
0696 3AEC37     LD   A,(37ECH)
```

```
0699 3C        INC  A
069A FE02      CP   02H
069C DA7500    JP   C,0075H
069F 3E01      LD   A,01H
06A1 32E1v7    LD   (37E1H),A
06A4 21EC37    LD   HL,37ECH
06A7 11EF37    LD   DE,37EFH
06AA 3603      LD   (HL),03H
06AC 010000    LD   BC,0000H
06AF CD6000    CALL 0060H
06B2 CB46      BIT  00H,(HL)
06B4 20FC      JR   NZ,06B2H
06B6 AF        XOR  A
06B7 32EE37    LD   (37EEH),A
06BA 010042    LD   BC,4200H
06BD 3E8C      LD   A,8CH
06BF 77        LD   (HL),A
06C0 CB4E      BIT  01H,(HL)
06C2 28FC      JR   Z,06C0H
06C4 1A        LD   A,(DE)
06C5 02        LD   (BC),A
06C6 0C        INC  C
06C7 20F7      JR   NZ,06C0H
06C9 C30042    JP   4200H
06CC 01181A    LD   BC,1A18H
06CF C3AE19    JP   19AEH
06D2 C3961C    JP   1C96H
06D5 C3781D    JP   1D78H
06D8 C3901C    JP   1C90H
06DB C3D925    JP   25D9H
06DE C9        RET
06DF 00        NOP
06E0 00
06E1 C9        RET
06E2 00        NOP
06E3 00        NOP
06E4 FB        EI
06E5 C9        RET
06E6 00        NOP
06E7 01E303    LD   BC,03E3H
06EA 00        NOP
06EB 00        NOP
06EC 00        NOP
06ED 4B        LD   C,E
06EE 49        LD   C,C
06EF 07        RLCA
06F0 58        LD   E,B
06F1 04        INC  B
06F2 00        NOP
06F3 3C        INC  A
06F4 00
06F5 44        LD   B,H
06F6 4F        LD   C,A
06F7 068D      LD   B,8DH
06F9 05        DEC  B
06FA 43        LD   B,E
06FB 00        NOP
06FC 00        NOP
06FD 50        LD   D,B
06FE 52        LD   D,D
06FF C30050    JP   5000H
0702 C7        RST  00H
0703
0704
0705 3E00      LD   A,00H
0707 C9        RET
0708 218013    LD   HL,1380H
070B CDC209    CALL 09C2H
070E 1806      JR   0716H
0710 CDC209    CALL 09C2H
0713 CD8209    CALL 0982H
0716 78        LD   A,B
0717 B7        OR   A
0718 C8        RET  Z
0719 3A2441    LD   A,(4124H)
071C B7        OR   A
071D CAB409    JP   Z,09B4H
0720 90        SUB  B
0721 300C      JR   NC,072FH
0723 2F        CPL
0724 3C        INC  A
0725 EB        EX   DE,HL
0726 CDA409    CALL 09A4H
0729 EB        EX   DE,HL
072A CDB409    CALL 09B4H
072D C1        POP  BC
072E D1        POP  DE
072F FE19      CP   19H
0731 D0        RET  NC
0732 F5        PUSH AF
0733 CDDF09    CALL 09DFH
0736 67        LD   H,A
0737 F1        POP  AF
0738 CDD707    CALL 07D7H
073B B4        OR   H
073C 212141    LD   HL,4121H
073F F25407    JP   P,0754H
0742 CDB707    CALL 07B7H
0745 D29607    JP   NC,0796H
0748 23        INC  HL
0749 34        INC  (HL)
074A CAB207    JP   Z,07B2H
074D 2E01      LD   L,01H
074F CDEB07    CALL 07EBH
0752 1842      JR   0796H
0754 AF        XOR  A
0755 90        SUB  B
0756 47        LD   B,A
0757 7E        LD   A,(HL)
0758 9B        SBC  A,E
0759 5F        LD   E,A
075A 23        INC  HL
075B 7E        LD   A,(HL)
075C 9A        SBC  A,D
075D 57        LD   D,A
075E 23        INC  HL
075F 7E        LD   A,(HL)
0760 99        SBC  A,C
0761 4F        LD   C,A
0762 DCC307    CALL C,07C3H
```

```
0765 68       LD   L,B
0766 63       LD   H,E
0767 AF       XOR  A
0768 47       LD   B,A
0769 79       LD   A,C
076A B7       OR   A
076B 2018     JR   NZ,0785H
076D 4A       LD   C,D
076E 54       LD   D,H
076F 65       LD   H,L
0770 6F       LD   L,A
0771 78       LD   A,B
0772 D608     SUB  08H
0774 FEE0     CP   0E0H
0776 20F0     JR   NZ,0768H
0778 AF       XOR  A
0779 322441   LD   (4124H),A
077C C9       RET
077D 05       DEC  B
077E 29       ADD  HL,HL
077F 7A       LD   A,D
0780 17       RLA
0781 57       LD   D,A
0782 79       LD   A,C
0783 8F       ADC  A,A
0784 4F       LD   C,A
0785 F27D07   JP   P,077DH
0788 78       LD   A,B
0789 5C       LD   E,H
078A 45       LD   B,L
078B B7       OR   A
078C 2808     JR   Z,0796H
078E 212441   LD   HL,4124H
0791 86       ADD  A,(HL)
0792 77       LD   (HL),A
0793 30E3     JR   NC,0778H
0795 C8       RET  Z
0796 78       LD   A,B
0797 212441   LD   HL,4124H
079A B7       OR   A
079B FCA807   CALL M,07A8H
079E 46       LD   B,(HL)
079F 23       INC  HL
07A0 7E       LD   A,(HL)
07A1 E680     AND  80H
07A3 A9       XOR  C
07A4 4F       LD   C,A
07A5 C3B409   JP   09B4H
07A8 1C       INC  E
07A9
07AA 14       INC  D
07AB C0       RET  NZ
07AC 0C       INC  C
07AD
07AE 0E80     LD   C,80H
07B0 34       INC  (HL)
07B1 C0       RET  NZ
07B2 1E0A     LD   E,0AH
07B4 C3A219   JP   19A2H
07B7 7E       LD   A,(HL)
07B8 83       ADD  A,E
07B9 5F       LD   E,A
07BA 23       INC  HL
07BB 7E       LD   A,(HL)
07BC 8A       ADC  A,D
07BD 57       LD   D,A
07BE 23       INC  HL
07BF 7E       LD   A,(HL)
07C0 89       ADC  A,C
07C1 4F       LD   C,A
07C2
07C3 212541   LD   HL,4125H
07C6 7E       LD   A,(HL)
07C7 2F       CPL
07C8 77       LD   (HL),A
07C9 AF       XOR  A
07CA 6F       LD   L,A
07CB 90       SUB  B
07CC 47       LD   B,A
07CD 7D       LD   A,L
07CE 9B       SBC  A,E
07CF 5F       LD   E,A
07D0 7D       LD   A,L
07D1 9A       SBC  A,D
07D2 57       LD   D,A
07D3 7D       LD   A,L
07D4 99       SBC  A,C
07D5 4F       LD   C,A
07D6
07D7 0600     LD   B,00H
07D9 D608     SUB  08H
07DB 3807     JR   C,07E4H
07DD 43       LD   B,E
07DE 5A       LD   E,D
07DF 51       LD   D,C
07E0 0E00     LD   C,00H
07E2 18F5     JR   07D9H
07E4 C609     ADD  09H
07E6 6F       LD   L,A
07E7 AF       XOR  A
07E8 2D       DEC  L
07E9 C8       RET  Z
07EA 79       LD   A,C
07EB 1F       RRA
07EC 4F       LD   C,A
07ED 7A       LD   A,D
07EE 1F       RRA
07EF 57       LD   D,A
07F0 7B       LD   A,E
07F1 1F       RRA
07F2 5F       LD   E,A
07F3 78       LD   A,B
07F4 1F       RRA
07F5 47       LD   B,A
07F6 18EF     JR   07E7H
07F8 00       NOP
07F9 00       NOP
07FA 00       NOP
07FB 81       ADD  A,C
07FC 03       INC  BC
```

```
07FD AA          XOR  D
07FE 56          LD   D,(HL)
07FF 19          ADD  HL,DE
0800 80          ADD  A,B
0801 F1          POP  AF
0802 227680      LD   (8076H),HL
0805 45          LD   B,L
0806 AA          XOR  D
0807 3882        JR   C,078BH
0809 CD5509      CALL 0955H
080C
080D EA4A1E      JP   PE,1E4AH
0810 212441      LD   HL,4124H
0813 7E          LD   A,(HL)
0814 013580      LD   BC,8035H
0817 11F304      LD   DE,04F3H
081A 90          SUB  B
081B F5          PUSH AF
081C 70          LD   (HL),B
081D D5          PUSH DE
081E C5          PUSH BC
081F CD1607      CALL 0716H
0822 C1          POP  BC
0823 D1          POP  DE
0824 04          INC  B
0825 CDA208      CALL 08A2H
0828 21F807      LD   HL,07F8H
082B CD1007      CALL 0710H
082E 21FC07      LD   HL,07FCH
0831 CD9A14      CALL 149AH
0834 018080      LD   BC,8080H
0837 110000      LD   DE,0000H
083A CD1607      CALL 0716H
083D F1          POP  AF
083E CD890F      CALL 0F89H
0841 013180      LD   BC,8031H
0844 111872      LD   DE,7218H
0847 CD5509      CALL 0955H
084A C8          RET  Z
084B 2E00        LD   L,00H
084D CD1o09      CALL 0914H
0850 79          LD   A,C
0851 324F41      LD   (414FH),A
0854 EB          EX   DE,HL
0855 225041      LD   (4150H),HL
0858 010000      LD   BC,0000H
085B 50          LD   D,B
085C 58          LD   E,B
085D 216507      LD   HL,0765H
0860 E5          PUSH HL
0861 216908      LD   HL,0869H
0864 E5          PUSH HL
0865 E5          PUSH HL
0866 212141      LD   HL,4121H
0869 7E          LD   A,(HL)
086A 23          INC  HL
086B
086C 2824        JR   Z,0892H
086E E5          PUSH HL
086F 2E08        LD   L,08H
0871
0872 67          LD   H,A
0873 79          LD   A,C
0874 300B        JR   NC,0881H
0876 E5          PUSH HL
0877 2A5041      LD   HL,(4150H)
087A 19          ADD  HL,DE
087B EB          EX   DE,HL
087C E1          POP  HL
087D 3A4F41      LD   A,(414FH)
0880 89          ADC  A,C
0881 1F          RRA
0882 4F          LD   C,A
0883 7A          LD   A,D
0884 1F          RRA
0885 57          LD   D,A
0886 7B          LD   A,E
0887
0888 5F          LD   E,A
0889 78          LD   A,B
088A 1F          RRA
088B 47          LD   B,A
088C 2D          DEC  L
088D 7C          LD   A,H
088E 20E1        JR   NZ,0871H
0890 E1          POP  HL
0891 C9          RET
0892 43          LD   B,E
0893 5A          LD   E,D
0894 51          LD   D,C
0895 4F          LD   C,A
0896 C9          RET
0897 CDA409      CALL 09A4H
089A 21D80D      LD   HL,0DD8H
089D CDB109      CALL 09B1H
08A0 C1          POP  BC
08A1 D1          POP  DE
08A2 CD5509      CALL 0955H
08A5 CA9A19      JP   Z,199AH
08A8 2EFF        LD   L,0FFH
08AA CD1409      CALL 0914H
08AD 34          INC  (HL)
08AE 34          INC  (HL)
08AF 2B          DEC  HL
08B0 7E          LD   A,(HL)
08B1 328940      LD   (4089H),A
08B4 2B          DEC  HL
08B5 7E          LD   A,(HL)
08B6 328540      LD   (4085H),A
08B9 2B          DEC  HL
08BA 7E          LD   A,(HL)
08BB 328140      LD   (4081H),A
08BE 41          LD   B,C
08BF EB          EX   DE,HL
08C0 AF          XOR  A
08C1 4F          LD   C,A
08C2 57          LD   D,A
08C3 5F          LD   E,A
08C4 328C40      LD   (408CH),A
08C7 E5          PUSH HL
```

```
08C8 C5        PUSH BC
08C9 7D        LD   A,L
08CA CD8040    CALL 4080H
08CD DE00      SBC  00H
08CF 3F        CCF
08D0 3007      JR   NC,08D9H
08D2 328C40    LD   (408CH),A
08D5 F1        POP  AF
08D6 F1        POP  AF
08D7 37        SCF
08D8 D2C1E1    JP   NC,0E1C1H
08DB 79        LD   A,C
08DC 3C        INC  A
08DD 3D        DEC  A
08DE
08DF FA9707    JP   M,0797H
08E2 17        RLA
08E3 7B        LD   A,E
08E4 17        RLA
08E5 5F        LD   E,A
08E6 7A        LD   A,D
08E7 17        RLA
08E8 57        LD   D,A
08E9 79        LD   A,C
08EA 17        RLA
08EB 4F        LD   C,A
08EC 29        ADD  HL,HL
08ED 78        LD   A,B
08EE 17        RLA
08EF 47        LD   B,A
08F0 3A8C40    LD   A,(408CH)
08F3 17        RLA
08F4 328C40    LD   (408CH),A
08F7 79        LD   A,C
08F8 B2        OR   D
08F9 B3        OR   E
08FA 20CB      JR   NZ,08C7H
08FC E5        PUSH HL
08FD 212441    LD   HL,4124H
0900 35        DEC  (HL)
0901 E1        POP  HL
0902 20C3      JR   NZ,08C7H
0904 C3B207    JP   07B2H
0907 3EFF      LD   A,0FFH
0909 2EAF      LD   L,0AFH
090B 212D41    LD   HL,412DH
090E 4E        LD   C,(HL)
090F 23        INC  HL
0910 AE        XOR  (HL)
0911 47        LD   B,A
0912 2E00      LD   L,00H
0914 78        LD   A,B
0915 B7        OR   A
0916 281F      JR   Z,0937H
0918 7D        LD   A,L
0919 212441    LD   HL,4124H
091C AE        XOR  (HL)
091D 80        ADD  A,B
091E 47        LD   B,A
091F
```

```
0920 A8        XOR  B      PAGE 33
0921 78        LD   A,B
0922 F23609    JP   P,0936H
0925 C680      ADD  80H
0927 77        LD   (HL),A
0928 CA9008    JP   Z,0890H
092B CDDF09    CALL 09DFH
092E 77        LD   (HL),A
092F 2B        DEC  HL
0930
0931 CD5509    CALL 0955H
0934 2F        CPL
0935 E1        POP  HL
0936 B7        OR   A
0937 E1        POP  HL
0938 F27807    JP   P,0778H
093B C3B207    JP   07B2H
093E CDBF09    CALL 09BFH
0941 78        LD   A,B
0942 B7        OR   A
0943 C8        RET  Z
0944 C602      ADD  02H
0946 DAB207    JP   C,07B2H
0949 47        LD   B,A
094A CD1607    CALL 0716H
094D 212441    LD   HL,4124H
0950 34        INC  (HL)
0951 C0        RET  NZ
0952 C3B207    JP   07B2H
0955 3A2441    LD   A,(4124H)
0958 B7        OR   A
0959 C8        RET  Z
095A 3A2341    LD   A,(4123H)
095D FE2F      CP   2FH
095F 17        RLA
0960 9F        SBC  A,A
0961 C0        RET  NZ
0962 3C        INC  A
0963 C9        RET
0964 0688      LD   B,88H
0966 110000    LD   DE,0000H
0969 212441    LD   HL,4124H
096C 4F        LD   C,A
096D 70        LD   (HL),B
096E 0600      LD   B,00H
0970 23        INC  HL
0971 3680      LD   (HL),80H
0973 17        RLA
0974 C36207    JP   0762H
0977 CD9409    CALL 0994H
097A F0        RET  P
097B E7        RST  20H
097C FA5B0C    JP   M,0C5BH
097F CAF60A    JP   Z,0AF6H
0982 212341    LD   HL,4123H
0985 7E        LD   A,(HL)
0986 EE80      XOR  80H
0988 77        LD   (HL),A
0989
098A CD9409    CALL 0994H
```

```
098D 6F       LD   L,A
098E 17       RLA
098F 9F       SBC  A,A
0990 67       LD   H,A
0991 C39A0A   JP   0A9AH
0994 E7       RST  20H
0995 CAF60A   JP   Z,0AF6H
0998 F25509   JP   P,0955H
099B 2A2141   LD   HL,(4121H)
099E 7C       LD   A,H
099F B5       OR   L
09A0 C8       RET  Z
09A1 7C       LD   A,H
09A2 18BB     JR   095FH
09A4 EB       EX   DE,HL
09A5 2A2141   LD   HL,(4121H)
09A8 E3       EX   (SP),HL
09A9 E5       PUSH HL
09AA 2A2341   LD   HL,(4123H)
09AD E3       EX   (SP),HL
09AE E5       PUSH HL
09AF EB       EX   DE,HL
09B0
09B1 CDC209   CALL 09C2H
09B4 EB       EX   DE,HL
09B5 222141   LD   (4121H),HL
09B8 60       LD   H,B
09B9 69       LD   L,C
09BA 222341   LD   (4123H),HL
09BD EB       EX   DE,HL
09BE C9       RET
09BF 212141   LD   HL,4121H
09C2 5E       LD   E,(HL)
09C3 23       INC  HL
09C4 56       LD   D,(HL)
09C5 23       INC  HL
09C6 4E       LD   C,(HL)
09C7 23       INC  HL
09C8 46       LD   B,(HL)
09C9 23       INC  HL
09CA C9       RET
09CB 112141   LD   DE,4121H
09CE 0604     LD   B,04H
09D0 1805     JR   09D7H
09D2 EB       EX   DE,HL
09D3 3AAF40   LD   A,(40AFH)
09D6 47       LD   B,A
09D7 1A       LD   A,(DE)
09D8 77       LD   (HL),A
09D9 13       INC  DE
09DA 23       INC  HL
09DB 05       DEC  B
09DC 20F9     JR   NZ,09D7H
09DE
09DF 212341   LD   HL,4123H
09E2 7E       LD   A,(HL)
09E3 07       RLCA
09E4 37       SCF
09E5 1F       RRA
09E6 77       LD   (HL),A
09E7 3F       CCF
09E8 1F       RRA
09E9 23       INC  HL
09EA 23       INC  HL
09EB 77       LD   (HL),A
09EC 79       LD   A,C
09ED 07       RLCA
09EE 37       SCF
09EF
09F0 4F       LD   C,A
09F1 1F       RRA
09F2 AE       XOR  (HL)
09F3 C9       RET
09F4 212741   LD   HL,4127H
09F7 11D209   LD   DE,09D2H
09FA 1806     JR   0A02H
09FC 212741   LD   HL,4127H
09FF 11D309   LD   DE,09D3H
0A02 D5       PUSH DE
0A03 112141   LD   DE,4121H
0A06 E7       RST  20H
0A07 D8       RET  C
0A08 111D41   LD   DE,411DH
0A0B C9       RET
0A0C 78       LD   A,B
0A0D B7       OR   A
0A0E CA5509   JP   Z,0955H
0A11 215E09   LD   HL,095EH
0A14 E5       PUSH HL
0A15 CD5509   CALL 0955H
0A18 79       LD   A,C
0A19 C8       RET  Z
0A1A 212341   LD   HL,4123H
0A1D AE       XOR  (HL)
0A1E 79       LD   A,C
0A1F F8       RET  M
0A20 CD260A   CALL 0A26H
0A23 1F       RRA
0A24 A9       XOR  C
0A25 C9       RET
0A26 23       INC  HL
0A27 78       LD   A,B
0A28 BE       CP   (HL)
0A29 C0       RET  NZ
0A2A 2B       DEC  HL
0A2B 79       LD   A,C
0A2C BE       CP   (HL)
0A2D C0       RET  NZ
0A2E 2B       DEC  HL
0A2F 7A       LD   A,D
0A30 BE       CP   (HL)
0A31 C0       RET  NZ
0A32 2B       DEC  HL
0A33 7B       LD   A,E
0A34 96       SUB  (HL)
0A35 C0       RET  NZ
0A36 E1       POP  HL
0A37 E1       POP  HL
0A38
0A39 7A       LD   A,D
```

```
0A3A AC        XOR  H
0A3B 7C        LD   A,H
0A3C FA5F09    JP   M,095FH
0A3F BA        CP   D
0A40 C26009    JP   NZ,0960H
0A43 7D        LD   A,L
0A44 93        SUB  E
0A45 C26009    JP   NZ,0960H
0A48
0A49 212741    LD   HL,4127H
0A4C CDD309    CALL 09D3H
0A4F 112E41    LD   DE,412EH
0A52 1A        LD   A,(DE)
0A53 B7        OR   A
0A54 CA5509    JP   Z,0955H
0A57 215E09    LD   HL,095EH
0A5A E5        PUSH HL
0A5B CD5509    CALL 0955H
0A5E 1B        DEC  DE
0A5F 1A        LD   A,(DE)
0A60 4F        LD   C,A
0A61 C8        RET  Z
0A62 212341    LD   HL,4123H
0A65 AE        XOR  (HL)
0A66 79        LD   A,C
0A67 F8        RET  M
0A68 13        INC  DE
0A69 23        INC  HL
0A6A 0608      LD   B,08H
0A6C 1A        LD   A,(DE)
0A6D 96        SUB  (HL)
0A6E C2230A    JP   NZ,0A23H
0A71 1B        DEC  DE
0A72 2B        DEC  HL
0A73 05        DEC  B
0A74 20F6      JR   NZ,0A6CH
0A76 C1        POP  BC
0A77
0A78 CD4F0A    CALL 0A4FH
0A7B C25E09    JP   NZ,095EH
0A7E C9        RET
0A7F E7        RST  20H
0A80 2A2141    LD   HL,(4121H)
0A83 F8        RET  M
0A84 CAF60A    JP   Z,0AF6H
0A87 D4B90A    CALL NC,0AB9H
0A8A 21B207    LD   HL,07B2H
0A8D E5        PUSH HL
0A8E 3A2441    LD   A,(4124H)
0A91 FE90      CP   90H
0A93 300E      JR   NC,0AA3H
0A95 CDFB0A    CALL 0AFBH
0A98 EB        EX   DE,HL
0A99 D1        POP  DE
0A9A 222141    LD   (4121H),HL
0A9D 3E02      LD   A,02H
0A9F 32AF40    LD   (40AFH),A
0AA2 C9        RET
0AA3 018090    LD   BC,9080H
0AA6 110000    LD   DE,0000H
0AA9 CD0C0A    CALL 0A0CH
0AAC C0        RET  NZ
0AAD 61        LD   H,C
0AAE 6A        LD   L,D
0AAF 18E8      JR   0A99H
0AB1 E7        RST  20H
0AB2 E0        RET  PO
0AB3 FACC0A    JP   M,0ACCH
0AB6 CAF60A    JP   Z,0AF6H
0AB9 CDBF09    CALL 09BFH
0ABC CDEF0A    CALL 0AEFH
0ABF 78        LD   A,B
0AC0
0AC1 C8        RET  Z
0AC2 CDDF09    CALL 09DFH
0AC5 212041    LD   HL,4120H
0AC8 46        LD   B,(HL)
0AC9 C39607    JP   0796H
0ACC 2A2141    LD   HL,(4121H)
0ACF CDEF0A    CALL 0AEFH
0AD2 7C        LD   A,H
0AD3 55        LD   D,L
0AD4 1E00      LD   E,00H
0AD6 0690      LD   B,90H
0AD8 C36909    JP   0969H
0ADB E7        RST  20H
0ADC D0        RET  NC
0ADD CAF60A    JP   Z,0AF6H
0AE0 FCCC0A    CALL M,0ACCH
0AE3 210000    LD   HL,0000H
0AE6 221D41    LD   (411DH),HL
0AE9 221F41    LD   (411FH),HL
0AEC 3E08      LD   A,08H
0AEE 013E04    LD   BC,043EH
0AF1 C39F0A    JP   0A9FH
0AF4 E7        RST  20H
0AF5 C8        RET  Z
0AF6 1E18      LD   E,18H
0AF8 C3A219    JP   19A2H
0AFB 47        LD   B,A
0AFC 4F        LD   C,A
0AFD 57        LD   D,A
0AFE 5F        LD   E,A
0AFF
0B00 C8        RET  Z
0B01 E5        PUSH HL
0B02 CDBF09    CALL 09BFH
0B05 CDDF09    CALL 09DFH
0B08 AE        XOR  (HL)
0B09 67        LD   H,A
0B0A FC1F0B    CALL M,0B1FH
0B0D 3E98      LD   A,98H
0B0F 90        SUB  B
0B10 CDD707    CALL 07D7H
0B13 7C        LD   A,H
0B14 17        RLA
0B15 DCA807    CALL C,07A8H
0B18 0600      LD   B,00H
0B1A DCC307    CALL C,07C3H
0B1D E1        POP  HL
```

```
0B1E
0B1F 1B        DEC  DE
0B20 7A        LD   A,D
0B21 A3        AND  E
0B22 3C        INC  A
0B23 C0        RET  NZ
0B24 0B        DEC  BC
0B25 C9        RET
0B26 E7        RST  20H
0B27 F8        RET  M
0B28 CD5509    CALL 0955H
0B2B F2370B    JP   P,0B37H
0B2E CD8209    CALL 0982H
0B31 CD370B    CALL 0B37H
0B34 C37B09    JP   097BH
0B37 E7        RST  20H
0B38 F8        RET  M
0B39 301E      JR   NC,0B59H
0B3B 28B9      JR   Z,0AF6H
0B3D CD8E0A    CALL 0A8EH
0B40 212441    LD   HL,4124H
0B43 7E        LD   A,(HL)
0B44 FE98      CP   98H
0B46 3A2141    LD   A,(4121H)
0B49 D0        RET  NC
0B4A 7E        LD   A,(HL)
0B4B CDFB0A    CALL 0AFBH
0B4E 3698      LD   (HL),98H
0B50 7B        LD   A,E
0B51 F5        PUSH AF
0B52 79        LD   A,C
0B53 17        RLA
0B54 CD6207    CALL 0762H
0B57 F1        POP  AF
0B58
0B59 212441    LD   HL,4124H
0B5C 7E        LD   A,(HL)
0B5D FE90      CP   90H
0B5F DA7F0A    JP   C,0A7FH
0B62 2014      JR   NZ,0B78H
0B64 4F        LD   C,A
0B65 2B        DEC  HL
0B66 7E        LD   A,(HL)
0B67 EE80      XOR  80H
0B69 0606      LD   B,06H
0B6B 2B        DEC  HL
0B6C B6        OR   (HL)
0B6D 05        DEC  B
0B6E 20FB      JR   NZ,0B6BH
0B70 B7        OR   A
0B71 210080    LD   HL,8000H
0B74 CA9A0A    JP   Z,0A9AH
0B77 79        LD   A,C
0B78 FEB8      CP   0B8H
0B7A D0        RET  NC
0B7B F5        PUSH AF
0B7C CDBF09    CALL 09BFH
0B7F CDDF09    CALL 09DFH
0B82 AE        XOR  (HL)
0B83 2B        DEC  HL
0B84 36B8      LD   (HL),0B8H
0B86 F5        PUSH AF
0B87 FCA00B    CALL M,0BA0H
0B8A 212341    LD   HL,4123H
0B8D 3EB8      LD   A,0B8H
0B8F 90        SUB  B
0B90 CD690D    CALL 0D69H
0B93 F1        POP  AF
0B94 FC200D    CALL M,0D20H
0B97 AF        XOR  A
0B98 321C41    LD   (411CH),A
0B9B F1        POP  AF
0B9C D0        RET  NC
0B9D C3D80C    JP   0CD8H
0BA0 211D41    LD   HL,411DH
0BA3 7E        LD   A,(HL)
0BA4 35        DEC  (HL)
0BA5 B7        OR   A
0BA6 23        INC  HL
0BA7 28FA      JR   Z,0BA3H
0BA9 C9        RET
0BAA
0BAB 210000    LD   HL,0000H
0BAE 78        LD   A,B
0BAF B1        OR   C
0BB0 2812      JR   Z,0BC4H
0BB2 3E10      LD   A,10H
0BB4 29        ADD  HL,HL
0BB5 DA3D27    JP   C,273DH
0BB8 EB        EX   DE,HL
0BB9 29        ADD  HL,HL
0BBA EB        EX   DE,HL
0BBB 3004      JR   NC,0BC1H
0BBD 09        ADD  HL,BC
0BBE DA3D27    JP   C,273DH
0BC1 3D        DEC  A
0BC2 20F0      JR   NZ,0BB4H
0BC4 EB        EX   DE,HL
0BC5 E1        POP  HL
0BC6 C9        RET
0BC7 7C        LD   A,H
0BC8 17        RLA
0BC9 9F        SBC  A,A
0BCA 47        LD   B,A
0BCB CD510C    CALL 0C51H
0BCE 79        LD   A,C
0BCF 98        SBC  A,B
0BD0 1803      JR   0BD5H
0BD2 7C        LD   A,H
0BD3 17        RLA
0BD4 9F        SBC  A,A
0BD5 47        LD   B,A
0BD6
0BD7 7A        LD   A,D
0BD8 17        RLA
0BD9 9F        SBC  A,A
0BDA 19        ADD  HL,DE
0BDB 88        ADC  A,B
0BDC 0F        RRCA
0BDD AC        XOR  H
```

```
0BDE F2990A     JP    P,0A99H
0BE1 C5         PUSH  BC
0BE2 EB         EX    DE,HL
0BE3 CDCF0A     CALL  0ACFH
0BE6 F1         POP   AF
0BE7 E1         POP   HL
0BE8 CDA409     CALL  09A4H
0BEB EB         EX    DE,HL
0BEC CD6B0C     CALL  0C6BH
0BEF C38F0F     JP    0F8FH
0BF2 7C         LD    A,H
0BF3 B5         OR    L
0BF4 CA9A0A     JP    Z,0A9AH
0BF7 E5         PUSH  HL
0BF8 D5         PUSH  DE
0BF9 CD450C     CALL  0C45H
0BFC C5         PUSH  BC
0BFD 44         LD    B,H
0BFE 4D         LD    C,L
0BFF 210000     LD    HL,0000H
0C02 3E10       LD    A,10H
0C04 29         ADD   HL,HL
0C05 381F       JR    C,0C26H
0C07 EB         EX    DE,HL
0C08 29         ADD   HL,HL
0C09 EB         EX    DE,HL
0C0A 3004       JR    NC,0C10H
0C0C 09         ADD   HL,BC
0C0D DA260C     JP    C,0C26H
0C10 3D         DEC   A
0C11 20F1       JR    NZ,0C04H
0C13
0C14 D1         POP   DE
0C15 7C         LD    A,H
0C16 B7         OR    A
0C17 FA1F0C     JP    M,0C1FH
0C1A D1         POP   DE
0C1B 78         LD    A,B
0C1C C34D0C     JP    0C4DH
0C1F EE80       XOR   80H
0C21 B5         OR    L
0C22 2813       JR    Z,0C37H
0C24 EB         EX    DE,HL
0C25 01C1E1     LD    BC,0E1C1H
0C28 CDCF0A     CALL  0ACFH
0C2B E1         POP   HL
0C2C CDA409     CALL  09A4H
0C2F CDCF0A     CALL  0ACFH
0C32 C1         POP   BC
0C33 D1         POP   DE
0C34 C34708     JP    0847H
0C3r 78         LD    A,B
0C38 B7         OR    A
0C39
0C3A FA9A0A     JP    M,0A9AH
0C3D D5         PUSH  DE
0C3E CDCF0A     CALL  0ACFH
0C41 D1         POP   DE
0C42 C38209     JP    0982H
0C45 7C         LD    A,H
0C46 AA         XOR   D
0C47 47         LD    B,A
0C48 CD4C0C     CALL  0C4CH
0C4B
0C4C 7C         LD    A,H
0C4D B7         OR    A
0C4E F29A0A     JP    P,0A9AH
0C51 AF         XOR   A
0C52 4F         LD    C,A
0C53 95         SUB   L
0C54 6F         LD    L,A
0C55 79         LD    A,C
0C56 9C         SBC   A,H
0C57 67         LD    H,A
0C58 C39A0A     JP    0A9AH
0C5B 2A2141     LD    HL,(4121H)
0C5E CD510C     CALL  0C51H
0C61 7C         LD    A,H
0C62 EE80       XOR   80H
0C64 B5         OR    L
0C65 C0         RET   NZ
0C66
0C67 CDEF0A     CALL  0AEFH
0C6A AF         XOR   A
0C6B 0698       LD    B,98H
0C6D C36909     JP    0969H
0C70 212D41     LD    HL,412DH
0C73 7E         LD    A,(HL)
0C74 EE80       XOR   80H
0C76 77         LD    (HL),A
0C77 212E41     LD    HL,412EH
0C7A 7E         LD    A,(HL)
0C7B B7         OR    A
0C7C C8         RET   Z
0C7D 47         LD    B,A
0C7E 2B         DEC   HL
0C7F 4E         LD    C,(HL)
0C80 112441     LD    DE,4124H
0C83 1A         LD    A,(DE)
0C84 B7         OR    A
0C85 CAF409     JP    Z,09F4H
0C88 90         SUB   B
0C89 3016       JR    NC,0CA1H
0C8B 2F         CPL
0C8C 3C         INC   A
0C8D F5         PUSH  AF
0C8E 0E08       LD    C,08H
0C90 23         INC   HL
0C91 E5         PUSH  HL
0C92 1A         LD    A,(DE)
0C93 46         LD    B,(HL)
0C94 77         LD    (HL),A
0C95 78         LD    A,B
0C96 12         LD    (DE),A
0C97 1B         DEC   DE
0C98 2B         DEC   HL
0C99 0D         DEC   C
0C9A 20F6       JR    NZ,0C92H
0C9C E1         POP   HL
0C9D 46         LD    B,(HL)
```

```
0C9E 2B       DEC  HL
0C9F 4E       LD   C,(HL)
0CA0 F1       POP  AF
0CA1 FE39     CP   39H
0CA3 D0       RET  NC
0CA4 F5       PUSH AF
0CA5 CDDF09   CALL 09DFH
0CA8
0CA9 3600     LD   (HL),00H
0CAB 47       LD   B,A
0CAC F1       POP  AF
0CAD 212D41   LD   HL,412DH
0CB0 CD690D   CALL 0D69H
0CB3 3A2641   LD   A,(4126H)
0CB6 321C41   LD   (411CH),A
0CB9 78       LD   A,B
0CBA B7       OR   A
0CBB F2CF0C   JP   P,0CCFH
0CBE CD330D   CALL 0D33H
0CC1 D20E0D   JP   NC,0D0EH
0CC4 EB       EX   DE,HL
0CC5 34       INC  (HL)
0CC6 CAB207   JP   Z,07B2H
0CC9 CD900D   CALL 0D90H
0CCC C30E0D   JP   0D0EH
0CCF CD450D   CALL 0D45H
0CD2 212541   LD   HL,4125H
0CD5 DC570D   CALL C,0D57H
0CD8 AF       XOR  A
0CD9 47       LD   B,A
0CDA 3A2341   LD   A,(4123H)
0CDD B7       OR   A
0CDE 201E     JR   NZ,0CFEH
0CE0 211C41   LD   HL,411CH
0CE3 0E08     LD   C,08H
0CE5 56       LD   D,(HL)
0CE6 77       LD   (HL),A
0CE7 7A       LD   A,D
0CE8
0CE9 0D       DEC  C
0CEA 20F9     JR   NZ,0CE5H
0CEC 78       LD   A,B
0CED D608     SUB  08H
0CEF FEC0     CP   0C0H
0CF1 20E6     JR   NZ,0CD9H
0CF3 C37807   JP   0778H
0CF6 05       DEC  B
0CF7 211C41   LD   HL,411CH
0CFA CD970D   CALL 0D97H
0CFD B7       OR   A
0CFE F2F60C   JP   P,0CF6H
0D01 78       LD   A,B
0D02 B7       OR   A
0D03 2809     JR   Z,0D0EH
0D05 212441   LD   HL,4124H
0D08 86       ADD  A,(HL)
0D09 77       LD   (HL),A
0D0A D27807   JP   NC,0778H
0D0D C8       RET  Z
0D0E 3A1C41   LD   A,(411CH)
0D11 B7       OR   A
0D12 FC200D   CALL M,0D20H
0D15 212541   LD   HL,4125H
0D18 7E       LD   A,(HL)
0D19 E680     AND  80H
0D1B 2B       DEC  HL
0D1C 2B       DEC  HL
0D1D AE       XOR  (HL)
0D1E 77       LD   (HL),A
0D1F C9       RET
0D20 211D41   LD   HL,411DH
0D23 0607     LD   B,07H
0D25 34       INC  (HL)
0D26 C0       RET  NZ
0D27 23       INC  HL
0D28 05       DEC  B
0D29 20FA     JR   NZ,0D25H
0D2B 34       INC  (HL)
0D2C CAB207   JP   Z,07B2H
0D2F 2B       DEC  HL
0D30 3680     LD   (HL),80H
0D32
0D33 212741   LD   HL,4127H
0D36 111D41   LD   DE,411DH
0D39 0E07     LD   C,07H
0D3B AF       XOR  A
0D3C 1A       LD   A,(DE)
0D3D 8E       ADC  A,(HL)
0D3E 12       LD   (DE),A
0D3F 13       INC  DE
0D40 23       INC  HL
0D41 0D       DEC  C
0D42 20F8     JR   NZ,0D3CH
0D44 C9       RET
0D45 212741   LD   HL,4127H
0D48 111D41   LD   DE,411DH
0D4B 0E07     LD   C,07H
0D4D AF       XOR  A
0D4E 1A       LD   A,(DE)
0D4F 9E       SBC  A,(HL)
0D50 12       LD   (DE),A
0D51 13       INC  DE
0D52 23       INC  HL
0D53 0D       DEC  C
0D54 20F8     JR   NZ,0D4EH
0D56
0D57 7E       LD   A,(HL)
0D58 2F       CPL
0D59 77       LD   (HL),A
0D5A 211C41   LD   HL,411CH
0D5D 0608     LD   B,08H
0D5F AF       XOR  A
0D60 4F       LD   C,A
0D61 79       LD   A,C
0D62 9E       SBC  A,(HL)
0D63 77       LD   (HL),A
0D64 23       INC  HL
0D65 05       DEC  B
0D66 20F9     JR   NZ,0D61H
0D68 C9       RET
```

```
0D69 71        LD   (HL),C
0D6A E5        PUSH HL
0D6B D608      SUB  08H
0D6D 380E      JR   C,0D7DH
0D6F E1        POP  HL
0D70 E5        PUSH HL
0D71 110008    LD   DE,0800H
0D74 4E        LD   C,(HL)
0D75 73        LD   (HL),E
0D76 59        LD   E,C
0D77 2B        DEC  HL
0D78
0D79 20F9      JR   NZ,0D74H
0D7B 18EE      JR   0D6BH
0D7D C609      ADD  09H
0D7F 57        LD   D,A
0D80 AF        XOR  A
0D81 E1        POP  HL
0D82
0D83 C8        RET  Z
0D84 E5        PUSH HL
0D85 1E08      LD   E,08H
0D87 7E        LD   A,(HL)
0D88 1F        RRA
0D89 77        LD   (HL),A
0D8A 2B        DEC  HL
0D8B 1D        DEC  E
0D8C 20F9      JR   NZ,0D87H
0D8E 18F0      JR   0D80H
0D90 212341    LD   HL,4123H
0D93 1601      LD   D,01H
0D95 18ED      JR   0D84H
0D97 0E08      LD   C,08H
0D99 7E        LD   A,(HL)
0D9A 17        RLA
0D9B 77        LD   (HL),A
0D9C 23        INC  HL
0D9D 0D        DEC  C
0D9E 20F9      JR   NZ,0D99H
0DA0 C9        RET
0DA1 CD5509    CALL 0955H
0DA4 C8        RET  Z
0DA5 CD0A09    CALL 090AH
0DA8 CD390E    CALL 0E39H
0DAB 71        LD   (HL),C
0DAC 13        INC  DE
0DAD 0607      LD   B,07H
0DAF 1A        LD   A,(DE)
0DB0 13        INC  DE
0DB1 B7        OR   A
0DB2 D5        PUSH DE
0DB3 2817      JR   Z,0DCCH
0DB5 0E08      LD   C,08H
0DB7 C5        PUSH BC
0DB8 1F        RRA
0DB9 47        LD   B,A
0DBA DC330D    CALL C,0D33H
0DBD CD900D    CALL 0D90H
0DC0 78        LD   A,B
0DC1 C1        POP  BC
0DC2 0D        DEC  C
0DC3 20F2      JR   NZ,0DB7H
0DC5
0DC6 05        DEC  B
0DC7 20E6      JR   NZ,0DAFH
0DC9 C3D80C    JP   0CD8H
0DCC 212341    LD   HL,4123H
0DCF CD700D    CALL 0D70H
0DD2 18F1      JR   0DC5H
0DD4 00        NOP
0DD5 00        NOP
0DD6 00        NOP
0DD7 00        NOP
0DD8 00        NOP
0DD9 00        NOP
0DDA 2084      JR   NZ,0D60H
0DDC 11D40D    LD   DE,0DD4H
0DDF 212741    LD   HL,4127H
0DE2 CDD309    CALL 09D3H
0DE5 3A2E41    LD   A,(412EH)
0DE8
0DE9 CA9A19    JP   Z,199AH
0DEC CD0709    CALL 0907H
0DEF 34        INC  (HL)
0DF0 34        INC  (HL)
0DF1 CD390E    CALL 0E39H
0DF4 215141    LD   HL,4151H
0DF7 71        LD   (HL),C
0DF8 41        LD   B,C
0DF9 114A41    LD   DE,414AH
0DFC 212741    LD   HL,4127H
0DFF CD4B0D    CALL 0D4BH
0E02 1A        LD   A,(DE)
0E03 99        SBC  A,C
0E04 3F        CCF
0E05 380B      JR   C,0E12H
0E07 114A41    LD   DE,414AH
0E0A 212741    LD   HL,4127H
0E0D CD390D    CALL 0D39H
0E10 AF        XOR  A
0E11 DA1204    JP   C,0412H
0E14 3A2341    LD   A,(4123H)
0E17 3C        INC  A
0E18 3D        DEC  A
0E19 1F        RRA
0E1A FA110D    JP   M,0D11H
0E1D 17        RLA
0E1E 211D41    LD   HL,411DH
0E21 0E07      LD   C,07H
0E23 CD990D    CALL 0D99H
0E26 214A41    LD   HL,414AH
0E29 CD970D    CALL 0D97H
0E2C 78        LD   A,B
0E2D B7        OR   A
0E2E 20C9      JR   NZ,0DF9H
0E30 212441    LD   HL,4124H
0E33 35        DEC  (HL)
0E34 20C3      JR   NZ,0DF9H
0E36 C3B207    JP   07B2H
0E39 79        LD   A,C
```

```
0E3A 322D41    LD   (412DH),A
0E3D 2B        DEC  HL
0E3E 115041    LD   DE,4150H
0E41 010007    LD   BC,0700H
0E44 7E        LD   A,(HL)
0E45 12        LD   (DE),A
0E46 71        LD   (HL),C
0E47 1B        DEC  DE
0E48 2B        DEC  HL
0E49 05        DEC  B
0E4A 20F8      JR   NZ,0E44H
0E4C C9        RET
0E4D CDFC09    CALL 09FCH
0E50 EB        EX   DE,HL
0E51 2B        DEC  HL
0E52 7E        LD   A,(HL)
0E53 B7        OR   A
0E54 C8        RET  Z
0E55 C602      ADD  02H
0E57 DAB207    JP   C,07B2H
0E5A 77        LD   (HL),A
0E5B
0E5C CD7r0C    CALL 0C77H
0E5F E1        POP  HL
0E60 34        INC  (HL)
0E61 C0        RET  NZ
0E62 C3B207    JP   07B2H
0E65 CD7807    CALL 0778H
0E68 CDEC0A    CALL 0AECH
0E6B F6AF      OR   0AFH
0E6D EB        EX   DE,HL
0E6E 01FF00    LD   BC,00FFH
0E71 60        LD   H,B
0E72 68        LD   L,B
0E73 CC9A0A    CALL Z,0A9AH
0E76 EB        EX   DE,HL
0E77 7E        LD   A,(HL)
0E78 FE2D      CP   2DH
0E7A F5        PUSH AF
0E7B CA830E    JP   Z,0E83H
0E7E FE2B      CP   2BH
0E80 2801      JR   Z,0E83H
0E82 2B        DEC  HL
0E83 D7        RST  10H
0E84 DA290F    JP   C,0F29H
0E87 FE2E      CP   2EH
0E89 CAE40E    JP   Z,0EE4H
0E8C FE45      CP   45H
0E8E 2814      JR   Z,0EA4H
0E90 FE25      CP   25H
0E92 CAEE0E    JP   Z,0EEEH
0E95 FE23      CP   23H
0E97 CAF50E    JP   Z,0EF5H
0E9A FE21      CP   21H
0E9C CAF60E    JP   Z,0EF6H
0E9F FE44      CP   44H
0EA1 2024      JR   NZ,0EC7H
0EA3 B7        OR   A
0EA4 CDFB0E    CALL 0EFBH
0EA7
0EA8 21BD0E    LD   HL,0EBDH
0EAB E3        EX   (SP),HL
0EAC D7        RST  10H
0EAD 15        DEC  D
0EAE FECE      CP   0CEH
0EB0 C8        RET  Z
0EB1 FE2D      CP   2DH
0EB3 C8        RET  Z
0EB4 14        INC  D
0EB5 FECD      CP   0CDH
0EB7 C8        RET  Z
0EB8 FE2B      CP   2BH
0EBA C8        RET  Z
0EBB 2B        DEC  HL
0EBC F1        POP  AF
0EBD D7        RST  10H
0EBE DA940F    JP   C,0F94H
0EC1 14        INC  D
0EC2 2003      JR   NZ,0EC7H
0EC4 AF        XOR  A
0EC5 93        SUB  E
0EC6 5F        LD   E,A
0EC7
0EC8 7B        LD   A,E
0EC9 90        SUB  B
0ECA F40A0F    CALL P,0F0AH
0ECD FC180F    CALL M,0F18H
0ED0 20F8      JR   NZ,0ECAH
0ED2 E1        POP  HL
0ED3 F1        POP  AF
0ED4 E5        PUSH HL
0ED5 CC7B09    CALL Z,097BH
0ED8 E1        POP  HL
0ED9 E7        RST  20H
0EDA E8        RET  PE
0EDB E5        PUSH HL
0EDC 219008    LD   HL,0890H
0EDF E5        PUSH HL
0EE0 CDA30A    CALL 0AA3H
0EE3
0EE4 E7        RST  20H
0EE5 0C        INC  C
0EE6 20DF      JR   NZ,0EC7H
0EE8 DCFB0E    CALL C,0EFBH
0EEB C3830E    JP   0E83H
0EEE E7        RST  20H
0EEF F29719    JP   P,1997H
0EF2 23        INC  HL
0EF3 18D2      JR   0EC7H
0EF5 B7        OR   A
0EF6 CDFB0E    CALL 0EFBH
0EF9 18F7      JR   0EF2H
0EFB E5        PUSH HL
0EFC D5        PUSH DE
0EFD C5        PUSH BC
0EFE F5        PUSH AF
0EFF CCB10A    CALL Z,0AB1H
0F02 F1        POP  AF
0F03 C4DB0A    CALL NZ,0ADBH
0F06 C1        POP  BC
```

```
0F07 D1          POP  DE
0F08 E1          POP  HL
0F09
0F0A C8          RET  Z
0F0B F5          PUSH AF
0F0C E7          RST  20H
0F0D F5          PUSH AF
0F0E E43E09      CALL PO,093EH
0F11 F1          POP  AF
0F12 EC4D0E      CALL PE,0E4DH
0F15 F1          POP  AF
0F16 3D          DEC  A
0F17
0F18 D5          PUSH DE
0F19 E5          PUSH HL
0F1A F5          PUSH AF
0F1B E7          RST  20H
0F1C F5          PUSH AF
0F1D E49708      CALL PO,0897H
0F20 F1          POP  AF
0F21 ECDC0D      CALL PE,0DDCH
0F24 F1          POP  AF
0F25 E1          POP  HL
0F26 D1          POP  DE
0F27 3C          INC  A
0F28 C9          RET
0F29 D5          PUSH DE
0F2A 78          LD   A,B
0F2B 89          ADC  A,C
0F2C 47          LD   B,A
0F2D C5          PUSH BC
0F2E E5          PUSH HL
0F2F 7E          LD   A,(HL)
0F30 D630        SUB  30H
0F32 F5          PUSH AF
0F33 E7          RST  20H
0F34 F25D0F      JP   P,0F5DH
0F37 2A2141      LD   HL,(4121H)
0F3A 11CD0C      LD   DE,0CCDH
0F3D DF          RST  18H
0F3E 3019        JR   NC,0F59H
0F40 54          LD   D,H
0F41 5D          LD   E,L
0F42 29          ADD  HL,HL
0F43 29          ADD  HL,HL
0F44 19          ADD  HL,DE
0F45 29          ADD  HL,HL
0F46 F1          POP  AF
0F47 4F          LD   C,A
0F48 09          ADD  HL,BC
0F49 7C          LD   A,H
0F4A B7          OR   A
0F4B FA570F      JP   M,0F57H
0F4E 222141      LD   (4121H),HL
0F51 E1          POP  HL
0F52 C1          POP  BC
0F53 D1          POP  DE
0F54 C3830E      JP   0E83H
0F57 79          LD   A,C
0F58 F5          PUSH AF
0F59 CDCC0A      CALL 0ACCH
0F5C 37          SCF
0F5D 3018        JR   NC,0F77H
0F5F 017494      LD   BC,9474H
0F62 110024      LD   DE,2400H
0F65 CD0C0A      CALL 0A0CH
0F68 F2740F      JP   P,0F74H
0F6B CD3E09      CALL 093EH
0F6E F1          POP  AF
0F6F CD890F      CALL 0F89H
0F72 18DD        JR   0F51H
0F74 CDE30A      CALL 0AE3H
0F77 CD4D0E      CALL 0E4DH
0F7A CDFC09      CALL 09FCH
0F7D F1          POP  AF
0F7E CD6409      CALL 0964H
0F81 CDE30A      CALL 0AE3H
0F84 CD770C      CALL 0C77H
0F87 18C8        JR   0F51H
0F89 CDA409      CALL 09A4H
0F8C CD6409      CALL 0964H
0F8F C1          POP  BC
0F90
0F91 C31607      JP   0716H
0F94 7B          LD   A,E
0F95 FE0A        CP   0AH
0F97 3009        JR   NC,0FA2H
0F99 07          RLCA
0F9A 07          RLCA
0F9B 83          ADD  A,E
0F9C 07          RLCA
0F9D 86          ADD  A,(HL)
0F9E D630        SUB  30H
0FA0 5F          LD   E,A
0FA1 FA1E32      JP   M,321EH
0FA4 C3BD0E      JP   0EBDH
0FA7 E5          PUSH HL
0FA8 212419      LD   HL,1924H
0FAB CDA728      CALL 28A7H
0FAE E1          POP  HL
0FAF CD9A0A      CALL 0A9AH
0FB2 AF          XOR  A
0FB3 CD3410      CALL 1034H
0FB6 B6          OR   (HL)
0FB7 CDD90F      CALL 0FD9H
0FBA C3A628      JP   28A6H
0FBD AF          XOR  A
0FBE CD3410      CALL 1034H
0FC1 E608        AND  08H
0FC3 2802        JR   Z,0FC7H
0FC5 362B        LD   (HL),2BH
0FC7 EB          EX   DE,HL
0FC8 CD9409      CALL 0994H
0FCB EB          EX   DE,HL
0FCC F2D90F      JP   P,0FD9H
0FCF 362D        LD   (HL),2DH
0FD1 C5          PUSH BC
0FD2 E5          PUSH HL
0FD3 CD7B09      CALL 097BH
0FD6
```

```
0FD7 C1        POP   BC
0FD8 B4        OR    H
0FD9 23        INC   HL
0FDA 3630      LD    (HL),30H
0FDC 3AD840    LD    A,(40D8H)
0FDF 57        LD    D,A
0FE0 17        RLA
0FE1 3AAF40    LD    A,(40AFH)
0FE4 DA9A10    JP    C,109AH
0FE7 CA9210    JP    Z,1092H
0FEA FE04      CP    04H
0FEC D23D10    JP    NC,103DH
0FEF 010000    LD    BC,0000H
0FF2 CD2F13    CALL  132FH
0FF5 213041    LD    HL,4130H
0FF8 46        LD    B,(HL)
0FF9 0E20      LD    C,20H
0FFB 3AD840    LD    A,(40D8H)
0FFE 5F        LD    E,A
0FFF E620      AND   20H
1001 2807      JR    Z,100AH
1003 78        LD    A,B
1004 B9        CP    C
1005 0E2A      LD    C,2AH
1007 2001      JR    NZ,100AH
1009 41        LD    B,C
100A 71        LD    (HL),C
100B D7        RST   10H
100C 2814      JR    Z,1022H
100E FE45      CP    45H
1010 2810      JR    Z,1022H
1012 FE44      CP    44H
1014 280C      JR    Z,1022H
1016 FE30      CP    30H
1018 28F0      JR    Z,100AH
101A FE2C      CP    2CH
101C 28EC      JR    Z,100AH
101E FE2E      CP    2EH
1020 2003      JR    NZ,1025H
1022 2B        DEC   HL
1023 3630      LD    (HL),30H
1025 7B        LD    A,E
1026 E610      AND   10H
1028 2803      JR    Z,102DH
102A 2B        DEC   HL
102B 3624      LD    (HL),24H
102D 7B        LD    A,E
102E E604      AND   04H
1030 C0        RET   NZ
1031 2B        DEC   HL
1032 70        LD    (HL),B
1033
1034 32D840    LD    (40D8H),A
1037 213041    LD    HL,4130H
103A 3620      LD    (HL),20H
103C
103D FE05      CP    05H
103F E5        PUSH  HL
1040 DE00      SBC   00H
1042 17        RLA
1043 57        LD    D,A
1044 14        INC   D
1045 CD0112    CALL  1201H
1048 010003    LD    BC,0300H
104B 82        ADD   A,D
104C FA5710    JP    M,1057H
104F 14        INC   D
1050 BA        CP    D
1051 3004      JR    NC,1057H
1053
1054 47        LD    B,A
1055 3E02      LD    A,02H
1057 D602      SUB   02H
1059 E1        POP   HL
105A F5        PUSH  AF
105B CD9112    CALL  1291H
105E 3630      LD    (HL),30H
1060 CCC909    CALL  Z,09C9H
1063 CDA412    CALL  12A4H
1066 2B        DEC   HL
1067 7E        LD    A,(HL)
1068 FE30      CP    30H
106A 28FA      JR    Z,1066H
106C FE2E      CP    2EH
106E C4C909    CALL  NZ,09C9H
1071 F1        POP   AF
1072 281F      JR    Z,1093H
1074 F5        PUSH  AF
1075 E7        RST   20H
1076 3E22      LD    A,22H
1078 8F        ADC   A,A
1079 77        LD    (HL),A
107A 23        INC   HL
107B F1        POP   AF
107C 362B      LD    (HL),2BH
107E F28510    JP    P,1085H
1081 362D      LD    (HL),2DH
1083 2F        CPL
1084
1085 062F      LD    B,2FH
1087 04        INC   B
1088 D60A      SUB   0AH
108A 30FB      JR    NC,1087H
108C C63A      ADD   3AH
108E 23        INC   HL
108F 70        LD    (HL),B
1090 23        INC   HL
1091 77        LD    (HL),A
1092 23        INC   HL
1093 3600      LD    (HL),00H
1095 EB        EX    DE,HL
1096 213041    LD    HL,4130H
1099 C9        RET
109A 23        INC   HL
109B C5        PUSH  BC
109C FE04      CP    04H
109E 7A        LD    A,D
109F D20911    JP    NC,1109H
10A2 1F        RRA
10A3 DAA311    JP    C,11A3H
```

```
10A6 010306    LD   BC,0603H
10A9 CD8912    CALL 1289H
10AC D1        POP  DE
10AD 7A        LD   A,D
10AE D605      SUB  05H
10B0 F46912    CALL P,1269H
10B3 CD2F13    CALL 132FH
10B6 7B        LD   A,E
10B7 B7        OR   A
10B8 CC2F09    CALL Z,092FH
10BB 3D        DEC  A
10BC F46912    CALL P,1269H
10BF E5        PUSH HL
10C0 CDF50F    CALL 0FF5H
10C3 E1        POP  HL
10C4 2802      JR   Z,10C8H
10C6 70        LD   (HL),B
10C7 23        INC  HL
10C8 3600      LD   (HL),00H
10CA 212F41    LD   HL,412FH
10CD 23        INC  HL
10CE 3AF340    LD   A,(40F3H)
10D1 95        SUB  L
10D2 92        SUB  D
10D3 C8        RET  Z
10D4 7E        LD   A,(HL)
10D5 FE20      CP   20H
10D7 28F4      JR   Z,10CDH
10D9 FE2A      CP   2AH
10DB 28F0      JR   Z,10CDH
10DD 2B        DEC  HL
10DE E5        PUSH HL
10DF F5        PUSH AF
10E0 01DF10    LD   BC,10DFH
10E3 C5        PUSH BC
10E4 D7        RST  10H
10E5 FE2D      CP   2DH
10E7
10E8 FE2B      CP   2BH
10EA C8        RET  Z
10EB FE24      CP   24H
10ED C8        RET  Z
10EE C1        POP  BC
10EF FE30      CP   30H
10F1 200F      JR   NZ,1102H
10F3 23        INC  HL
10F4 D7        RST  10H
10F5 300B      JR   NC,1102H
10F7 2B        DEC  HL
10F8 012B77    LD   BC,772BH
10FB F1        POP  AF
10FC 28FB      JR   Z,10F9H
10FE C1        POP  BC
10FF C3CE10    JP   10CEH
1102 F1        POP  AF
1103 28FD      JR   Z,1102H
1105 E1        POP  HL
1106 3625      LD   (HL),25H
1108
1109 E5        PUSH HL
110A 1F        RRA
110B DAAA11    JP   C,11AAH
110E 2814      JR   Z,1124H
1110 118413    LD   DE,1384H
1113 CD490A    CALL 0A49H
1116 1610      LD   D,10H
1118 FA3211    JP   M,1132H
111B E1        POP  HL
111C C1        POP  BC
111D CDBD0F    CALL 0FBDH
1120 2B        DEC  HL
1121 3625      LD   (HL),25H
1123
1124 010EB6    LD   BC,0B60EH
1127 11CA1B    LD   DE,1BCAH
112A CD0C0A    CALL 0A0CH
112D F21B11    JP   P,111BH
1130 1606      LD   D,06H
1132 CD5509    CALL 0955H
1135 C40112    CALL NZ,1201H
1138 E1        POP  HL
1139 C1        POP  BC
113A FA5711    JP   M,1157H
113D C5        PUSH BC
113E 5F        LD   E,A
113F 78        LD   A,B
1140 92        SUB  D
1141 93        SUB  E
1142 F46912    CALL P,1269H
1145 CD7D12    CALL 127DH
1148 CDA412    CALL 12A4H
114B B3        OR   E
114C C47712    CALL NZ,1277H
114F B3        OR   E
1150 C49112    CALL NZ,1291H
1153
1154 C3B610    JP   10B6H
1157 5F        LD   E,A
1158 79        LD   A,C
1159 B7        OR   A
115A C4160F    CALL NZ,0F16H
115D 83        ADD  A,E
115E FA6211    JP   M,1162H
1161 AF        XOR  A
1162 C5        PUSH BC
1163 F5        PUSH AF
1164 FC180F    CALL M,0F18H
1167 FA6411    JP   M,1164H
116A C1        POP  BC
116B 7B        LD   A,E
116C 90        SUB  B
116D C1        POP  BC
116E 5F        LD   E,A
116F 82        ADD  A,D
1170 78        LD   A,B
1171 FA7F11    JP   M,117FH
1174 92        SUB  D
1175 93        SUB  E
1176 F46912    CALL P,1269H
1179 C5        PUSH BC
```

```
117A CD7D12    CALL 127DH
117D 1811      JR   1190H
117F CD6912    CALL 1269H
1182 79        LD   A,C
1183 CD9412    CALL 1294H
1186 4F        LD   C,A
1187 AF        XOR  A
1188 92        SUB  D
1189 93        SUB  E
118A CD6912    CALL 1269H
118D C5        PUSH BC
118E 47        LD   B,A
118F 4F        LD   C,A
1190 CDA412    CALL 12A4H
1193 C1        POP  BC
1194 B1        OR   C
1195 2003      JR   NZ,119AH
1197 2AF340    LD   HL,(40F3H)
119A 83        ADD  A,E
119B 3D        DEC  A
119C F46912    CALL P,1269H
119F 50        LD   D,B
11A0 C3BF10    JP   10BFH
11A3 E5        PUSH HL
11A4 D5        PUSH DE
11A5 CDCC0A    CALL 0ACCH
11A8 D1        POP  DE
11A9 AF        XOR  A
11AA CAB011    JP   Z,11B0H
11AD 1E10      LD   E,10H
11AF 011E06    LD   BC,061EH
11B2 CD5509    CALL 0955H
11B5 37        SCF
11B6 C40112    CALL NZ,1201H
11B9
11BA C1        POP  BC
11BB F5        PUSH AF
11BC 79        LD   A,C
11BD           OR   A
11BE F5        PUSH AF
11BF C4160F    CALL NZ,0F16H
11C2 80        ADD  A,B
11C3 4F        LD   C,A
11C4 7A        LD   A,D
11C5 E604      AND  04H
11C7 FE01      CP   01H
11C9 9F        SBC  A,A
11CA 57        LD   D,A
11CB 81        ADD  A,C
11CC 4F        LD   C,A
11CD 93        SUB  E
11CE F5        PUSH AF
11CF C5        PUSH BC
11D0 FC180F    CALL M,0F18H
11D3 FAD011    JP   M,11D0H
11D6 C1        POP  BC
11D7 F1        POP  AF
11D8 C5        PUSH BC
11D9 F5        PUSH AF
11DA FADE11    JP   M,11DEH
11DD AF        XOR  A
11DE 2F        CPL
11DF 3C        INC  A
11E0 80        ADD  A,B
11E1 3C        INC  A
11E2 82        ADD  A,D
11E3 47        LD   B,A
11E4 0E00      LD   C,00H
11E6 CDA412    CALL 12A4H
11E9
11EA F47112    CALL P,1271H
11ED C1        POP  BC
11EE F1        POP  AF
11EF CC2F09    CALL Z,092FH
11F2 F1        POP  AF
11F3 3803      JR   C,11F8H
11F5 83        ADD  A,E
11F6 90        SUB  B
11F7 92        SUB  D
11F8 C5        PUSH BC
11F9 CD7410    CALL 1074H
11FC EB        EX   DE,HL
11FD D1        POP  DE
11FE C3BF10    JP   10BFH
1201 D5        PUSH DE
1202 AF        XOR  A
1203 F5        PUSH AF
1204 E7        RST  20H
1205 E22212    JP   PO,1222H
1208 3A2441    LD   A,(4124H)
120B FE91      CP   91H
120D D22212    JP   NC,1222H
1210 116413    LD   DE,1364H
1213 212741    LD   HL,4127H
1216 CDD309    CALL 09D3H
1219 CDA10D    CALL 0DA1H
121C           POP  AF
121D D60A      SUB  0AH
121F F5        PUSH AF
1220 18E6      JR   1208H
1222 CD4F12    CALL 124FH
1225 E7        RST  20H
1226 300B      JR   NC,1233H
1228 014391    LD   BC,9143H
122B 11F94F    LD   DE,4FF9H
122E CD0C0A    CALL 0A0CH
1231 1806      JR   1239H
1233 116C13    LD   DE,136CH
1236 CD490A    CALL 0A49H
1239 F24B12    JP   P,124BH
123C F1        POP  AF
123D CD0B0F    CALL 0F0BH
1240 F5        PUSH AF
1241 18E2      JR   1225H
1243 F1        POP  AF
1244 CD180F    CALL 0F18H
1247 F5        PUSH AF
1248 CD4F12    CALL 124FH
124B F1        POP  AF
124C
```

```
124D D1          POP   DE
124E C9          RET
124F E7          RST   20H
1250 EA5E12      JP    PE,125EH
1253 017494      LD    BC,9474H
1256 11F823      LD    DE,23F8H
1259 CD0C0A      CALL  0A0CH
125C 1806        JR    1264H
125E 117413      LD    DE,1374H
1261 CD490A      CALL  0A49H
1264 E1          POP   HL
1265 F24312      JP    P,1243H
1268 E9          JP    (HL)
1269 B7          OR    A
126A C8          RET   Z
126B 3D          DEC   A
126C 3630        LD    (HL),30H
126E 23          INC   HL
126F 18F9        JR    126AH
1271 2004        JR    NZ,1277H
1273 C8          RET   Z
1274 CD9112      CALL  1291H
1277 3630        LD    (HL),30H
1279 23          INC   HL
127A 3D          DEC   A
127B 18F6        JR    1273H
127D 7B          LD    A,E
127E 82          ADD   A,D
127F 3C          INC   A
1280 47          LD    B,A
1281 3C          INC   A
1282 D603        SUB   03H
1284 30FC        JR    NC,1282H
1286 C605        ADD   05H
1288 4F          LD    C,A
1289 3AD840      LD    A,(40D8H)
128C E640        AND   40H
128E C0          RET   NZ
128F 4F          LD    C,A
1290
1291 05          DEC   B
1292 2008        JR    NZ,129CH
1294 362E        LD    (HL),2EH
1296 22F340      LD    (40F3H),HL
1299 23          INC   HL
129A 48          LD    C,B
129B C9          RET
129C 0D          DEC   C
129D C0          RET   NZ
129E 362C        LD    (HL),2CH
12A0 23          INC   HL
12A1 0E03        LD    C,03H
12A3
12A4 D5          PUSH  DE
12A5 E7          RST   20H
12A6 E2EA12      JP    PO,12EAH
12A9 C5          PUSH  BC
12AA E5          PUSH  HL
12AB CDFC09      CALL  09FCH
12AE 217C13      LD    HL,137CH
12B1 CDF709      CALL  09F7H
12B4 CD770C      CALL  0C77H
12B7 AF          XOR   A
12B8 CD7B0B      CALL  0B7BH
12BB
12BC C1          POP   BC
12BD 118C13      LD    DE,138CH
12C0 3E0A        LD    A,0AH
12C2 CD9112      CALL  1291H
12C5 C5          PUSH  BC
12C6 F5          PUSH  AF
12C7 E5          PUSH  HL
12C8 D5          PUSH  DE
12C9 062F        LD    B,2FH
12CB 04          INC   B
12CC
12CD E5          PUSH  HL
12CE CD480D      CALL  0D48H
12D1 30F8        JR    NC,12CBH
12D3 E1          POP   HL
12D4 CD360D      CALL  0D36H
12D7 EB          EX    DE,HL
12D8 E1          POP   HL
12D9 70          LD    (HL),B
12DA 23          INC   HL
12DB F1          POP   AF
12DC C1          POP   BC
12DD 3D          DEC   A
12DE 20E2        JR    NZ,12C2H
12E0 C5          PUSH  BC
12E1 E5          PUSH  HL
12E2 211D41      LD    HL,411DH
12E5 CDB109      CALL  09B1H
12E8 180C        JR    12F6H
12EA C5          PUSH  BC
12EB E5          PUSH  HL
12EC CD0807      CALL  0708H
12EF 3C          INC   A
12F0 CDFB0A      CALL  0AFBH
12F3 CDB409      CALL  09B4H
12F6 E1          POP   HL
12F7 C1          POP   BC
12F8 AF          XOR   A
12F9 11D213      LD    DE,13D2H
12FC 3F          CCF
12FD CD9112      CALL  1291H
1300 C5          PUSH  BC
1301 F5          PUSH  AF
1302 E5          PUSH  HL
1303 D5          PUSH  DE
1304 CDBF09      CALL  09BFH
1307 E1          POP   HL
1308 062F        LD    B,2FH
130A 04          INC   B
130B 7B          LD    A,E
130C 96          SUB   (HL)
130D 5F          LD    E,A
130E 23          INC   HL
130F 7A          LD    A,D
1310 9E          SBC   A,(HL)
```

```
1311 57        LD   D,A
1312 23        INC  HL
1313 79        LD   A,C
1314 9E        SBC  A,(HL)
1315 4F        LD   C,A
1316 2B        DEC  HL
1317 2B        DEC  HL
1318 30F0      JR   NC,130AH
131A CDB707    CALL 07B7H
131D 23        INC  HL
131E CDB409    CALL 09B4H
1321 EB        EX   DE,HL
1322 E1        POP  HL
1323 70        LD   (HL),B
1324
1325 F1        POP  AF
1326 C1        POP  BC
1327 38D3      JR   C,12FCH
1329 13        INC  DE
132A 13        INC  DE
132B 3E04      LD   A,04H
132D 1806      JR   1335H
132F D5        PUSH DE
1330 11D813    LD   DE,13D8H
1333 3E05      LD   A,05H
1335 CD9112    CALL 1291H
1338 C5        PUSH BC
1339 F5        PUSH AF
133A E5        PUSH HL
133B EB        EX   DE,HL
133C 4E        LD   C,(HL)
133D 23        INC  HL
133E 46        LD   B,(HL)
133F C5        PUSH BC
1340
1341 E3        EX   (SP),HL
1342 EB        EX   DE,HL
1343 2A2141    LD   HL,(4121H)
1346 062F      LD   B,2FH
1348 04        INC  B
1349 7D        LD   A,L
134A 93        SUB  E
134B 6F        LD   L,A
134C 7C        LD   A,H
134D 9A        SBC  A,D
134E 67        LD   H,A
134F 30F7      JR   NC,1348H
1351 19        ADD  HL,DE
1352 222141    LD   (4121H),HL
1355 D1        POP  DE
1356 E1        POP  HL
1357 70        LD   (HL),B
1358 23        INC  HL
1359 F1        POP  AF
135A C1        POP  BC
135B 3D        DEC  A
135C 20D7      JR   NZ,1335H
135E CD9112    CALL 1291H
1361 77        LD   (HL),A
1362 D1        POP  DE
1363
1364 00        NOP
1365 00        NOP
1366 00        NOP
1367 00        NOP
1368 F9        LD   SP,HL
1369 02        LD   (BC),A
136A 15        DEC  D
136B A2        AND  D
136C FDFF      INDEX SPECIFIED
136E 9F        SBC  A,A
136F 31A95F    LD   SP,5FA9H
1372 63        LD   H,E
1373 B2        OR   D
1374 FEFF      CP   0FFH
1376 03        INC  BC
1377 BF        CP   A
1378
1379 1B        DEC  DE
137A 0EB6      LD   C,0B6H
137C 00        NOP
137D 00        NOP
137E 00        NOP
137F 00        NOP
1380 00        NOP
1381 00        NOP
1382 00        NOP
1383 80        ADD  A,B
1384 00        NOP
1385 00        NOP
1386 04        INC  B
1387 BF        CP   A
1388 C9        RET
1389 1B        DEC  DE
138A 0EB6      LD   C,0B6H
138C 00        NOP
138D 80        ADD  A,B
138E C6A4      ADD  0A4H
1390 7E        LD   A,(HL)
1391 8D        ADC  A,L
1392 03        INC  BC
1393 00        NOP
1394 40        LD   B,B
1395 7A        LD   A,D
1396 10F3      DJNZ 138BH
1398 5A        LD   E,D
1399 00        NOP
139A 00        NOP
139B A0        AND  B
139C 72        LD   (HL),D
139D 4E        LD   C,(HL)
139E 1809      JR   13A9H
13A0 00        NOP
13A1 00        NOP
13A2 10A5      DJNZ 1349H
13A4 D4E800    CALL NC,00E8H
13A7 00        NOP
13A8 00        NOP
13A9 E8        RET  PE
13AA 76        HALT
```

```
13AB 48       LD   C,B
13AC 17       RLA
13AD 00       NOP
13AE 00       NOP
13AF 00       NOP
13B0 E40B54   CALL PO,540BH
13B3 02       LD   (BC),A
13B4 00       NOP
13B5 00       NOP
13B6 00       NOP
13B7 CA9A3B   JP   Z,3B9AH
13BA 00       NOP
13BB 00       NOP
13BC 00       NOP
13BD 00       NOP
13BE E1       POP  HL
13BF F5       PUSH AF
13C0 05       DEC  B
13C1 00       NOP
13C2 00       NOP
13C3 00       NOP
13C4 80       ADD  A,B
13C5 96       SUB  (HL)
13C6 98       SBC  A,B
13C7 00       NOP
13C8 00       NOP
13C9 00       NOP
13CA 00       NOP
13CB 40       LD   B,B
13CC 42       LD   B,D
13CD 0F       RRCA
13CE 00       NOP
13CF 00       NOP
13D0 00       NOP
13D1 00       NOP
13D2 A0       AND  B
13D3 86       ADD  A,(HL)
13D4 011027   LD   BC,2710H
13D7 00       NOP
13D8 1027     DJNZ 1401H
13DA E8       RET  PE
13DB
13DC 64       LD   H,H
13DD 00       NOP
13DE 0A       LD   A,(BC)
13DF 00       NOP
13E0 010021   LD   BC,2100H
13E3 82       ADD  A,D
13E4 09       ADD  HL,BC
13E5 E3       EX   (SP),HL
13E6 E9       JP   (HL)
13E7 CDA409   CALL 09A4H
13EA 218013   LD   HL,1380H
13ED CDB109   CALL 09B1H
13F0 1803     JR   13F5H
13F2 CDB10A   CALL 0AB1H
13F5
13F6 D1       POP  DE
13F7 CD5509   CALL 0955H
13FA 78       LD   A,B
13FB 283C     JR   Z,1439H
13FD F20414   JP   P,1404H
1400 B7       OR   A
1401 CA9A19   JP   Z,199AH
1404 B7       OR   A
1405 CA7907   JP   Z,0779H
1408 D5       PUSH DE
1409 C5       PUSH BC
140A 79       LD   A,C
140B F67F     OR   7FH
140D CDBF09   CALL 09BFH
1410 F22114   JP   P,1421H
1413 D5       PUSH DE
1414 C5       PUSH BC
1415 CD400B   CALL 0B40H
1418 C1       POP  BC
1419 D1       POP  DE
141A F5       PUSH AF
141B CD0C0A   CALL 0A0CH
141E E1       POP  HL
141F 7C       LD   A,H
1420 1F       RRA
1421 E1       POP  HL
1422 222341   LD   (4123H),HL
1425 E1       POP  HL
1426 222141   LD   (4121H),HL
1429 DCE213   CALL C,13E2H
142C CC8209   CALL Z,0982H
142F D5       PUSH DE
1430
1431 CD0908   CALL 0809H
1434
1435 D1       POP  DE
1436 CD4708   CALL 0847H
1439 CDA409   CALL 09A4H
143C 013881   LD   BC,8138H
143F 113BAA   LD   DE,0AA3BH
1442 CD4708   CALL 0847H
1445 3A2441   LD   A,(4124H)
1448 FE88     CP   88H
144A D23109   JP   NC,0931H
144D CD400B   CALL 0B40H
1450 C680     ADD  80H
1452 C602     ADD  02H
1454 DA3109   JP   C,0931H
1457 F5       PUSH AF
1458 21F807   LD   HL,07F8H
145B CD0B07   CALL 070BH
145E CD4108   CALL 0841H
1461 F1       POP  AF
1462 C1       POP  BC
1463 D1       POP  DE
1464 F5       PUSH AF
1465 CD1307   CALL 0713H
1468 CD8209   CALL 0982H
146B 217914   LD   HL,1479H
146E CDA914   CALL 14A9H
1471 110000   LD   DE,0000H
1474 C1       POP  BC
1475 4A       LD   C,D
```

```
1476 C34708     JP    0847H
1479 08         EX    AF,AF'
147A 40         LD    B,B
147B 2E94       LD    L,94H
147D 74         LD    (HL),H
147E 70         LD    (HL),B
147F 4F         LD    C,A
1480 2E77       LD    L,77H
1482 6E         LD    L,(HL)
1483 02         LD    (BC),A
1484 88         ADC   A,B
1485 7A         LD    A,D
1486 E6A0       AND   0A0H
1488 2A7C50     LD    HL,(507CH)
148B AA         XOR   D
148C AA         XOR   D
148D 7E         LD    A,(HL)
148E FF         RST   38H
148F FF         RST   38H
1490 7F         LD    A,A
1491 7F         LD    A,A
1492 00         NOP
1493 00         NOP
1494 80         ADD   A,B
1495 81         ADD   A,C
1496 00         NOP
1497 00         NOP
1498 00         NOP
1499 81         ADD   A,C
149A CDA409     CALL  09A4H
149D 11320C     LD    DE,0C32H
14A0 D5         PUSH  DE
14A1 E5         PUSH  HL
14A2 CDBF09     CALL  09BFH
14A5 CD4708     CALL  0847H
14A8 E1         POP   HL
14A9 CDA409     CALL  09A4H
14AC 7E         LD    A,(HL)
14AD 23         INC   HL
14AE CDB109     CALL  09B1H
14B1 06F1       LD    B,0F1H
14B3 C1         POP   BC
14B4
14B5 3D         DEC   A
14B6 C8         RET   Z
14B7 D5         PUSH  DE
14B8 C5         PUSH  BC
14B9 F5         PUSH  AF
14BA E5         PUSH  HL
14BB CD4708     CALL  0847H
14BE E1         POP   HL
14BF CDC209     CALL  09C2H
14C2 E5         PUSH  HL
14C3 CD1607     CALL  0716H
14C6 E1         POP   HL
14C7 18E9       JR    14B2H
14C9 CD7F0A     CALL  0A7FH
14CC 7C         LD    A,H
14CD
14CE FA4A1E     JP    M,1E4AH
14D1 B5         OR    L
14D2 CAF014     JP    Z,14F0H
14D5 E5         PUSH  HL
14D6 CDF014     CALL  14F0H
14D9 CDBF09     CALL  09BFH
14DC

14DD E3         EX    (SP),HL
14DE C5         PUSH  BC
14DF CDCF0A     CALL  0ACFH
14E2 C1         POP   BC
14E3 D1         POP   DE
14E4 CD4708     CALL  0847H
14E7 21F807     LD    HL,07F8H
14EA CD0B07     CALL  070BH
14ED C3400B     JP    0B40H
14F0 219040     LD    HL,4090H
14F3 E5         PUSH  HL
14F4 110000     LD    DE,0000H
14F7 4B         LD    C,E
14F8 2603       LD    H,03H
14FA 2E08       LD    L,08H
14FC EB         EX    DE,HL
14FD 29         ADD   HL,HL
14FE EB         EX    DE,HL
14FF 79         LD    A,C
1500 17         RLA
1501 4F         LD    C,A
1502 E3         EX    (SP),HL
1503 7E         LD    A,(HL)
1504 07         RLCA
1505 77         LD    (HL),A
1506 E3         EX    (SP),HL
1507 D21615     JP    NC,1516H
150A E5         PUSH  HL
150B 2AAA40     LD    HL,(40AAH)
150E 19         ADD   HL,DE
150F EB         EX    DE,HL
1510 3AAC40     LD    A,(40ACH)
1513 89         ADC   A,C
1514 4F         LD    C,A
1515 E1         POP   HL
1516 2D         DEC   L
1517 C2FC14     JP    NZ,14FCH
151A E3         EX    (SP),HL
151B 23         INC   HL
151C E3         EX    (SP),HL
151D 25         DEC   H
151E C2FA14     JP    NZ,14FAH
1521 E1         POP   HL
1522 2165B0     LD    HL,0B065H
1525 19         ADD   HL,DE
1526 22AA40     LD    (40AAH),HL
1529 CDEF0A     CALL  0AEFH
152C 3E05       LD    A,05H
152E 89         ADC   A,C
152F 32AC40     LD    (40ACH),A
1532
1533 0680       LD    B,80H
1535 21.541     LD    HL,4125H
1538 70         LD    (HL),B
```

```
1539 2B        DEC  HL
153A 70        LD   (HL),B
153B 4F        LD   C,A
153C 0600      LD   B,00H
153E C36507    JP   0765H
1541 218B15    LD   HL,158BH
1544 CD0B07    CALL 070BH
1547 CDA409    CALL 09A4H
154A 014983    LD   BC,8349H
154D 11DB0F    LD   DE,0FDBH
1550 CDB409    CALL 09B4H
1553
1554
1555 CDA208    CALL 08A2H
1558 CDA409    CALL 09A4H
155B CD400B    CALL 0B40H
155E C1        POP  BC
155F D1        POP  DE
1560 CD1307    CALL 0713H
1563 218F15    LD   HL,158FH
1566 CD1007    CALL 0710H
1569 CD5509    CALL 0955H
156C 37        SCF
156D F27715    JP   P,1577H
1570 CD0807    CALL 0708H
1573 CD5509    CALL 0955H
1576 B7        OR   A
1577 F5        PUSH AF
1578 F48209    CALL P,0982H
157B 218F15    LD   HL,158FH
157E CD0B07    CALL 070BH
1581 F1        POP  AF
1582 D48209    CALL NC,0982H
1585 219315    LD   HL,1593H
1588 C39A14    JP   149AH
158B DB0F      IN   A,(0FH)
158D 49        LD   C,C
158E 81        ADD  A,C
158F 00        NOP
1590 00        NOP
1591 00        NOP
1592 7F        LD   A,A
1593 05        DEC  B
1594 BA        CP   D
1595 D7        RST  10H
1596 1E86      LD   E,86H
1598 64        LD   H,H
1599 2699      LD   H,99H
159B 87        ADD  A,A
159C 58        LD   E,B
159D 34        INC  (HL)
159E 23        INC  HL
159F 87        ADD  A,A
15A0 E0        RET  PO
15A1 5D        LD   E,L
15A2 A5        AND  L
15A3 86        ADD  A,(HL)
15A4 DA0F49    JP   C,490FH
15A7 83        ADD  A,E
15A8 CDA409    CALL 09A4H
15AB CD4715    CALL 1547H
15AE C1        POP  BC
15AF E1        POP  HL
15B0 CDA409    CALL 09A4H
15B3 EB        EX   DE,HL
15B4 CDB409    CALL 09B4H
15B7 CD4115    CALL 1541H
15BA C3A008    JP   08A0H
15BD CD5509    CALL 0955H
15C0 FCE213    CALL M,13E2H
15C3 FC8209    CALL M,0982H
15C6 3A2441    LD   A,(4124H)
15C9 FE81      CP   81H
15CB 380C      JR   C,15D9H
15CD 010081    LD   BC,8100H
15D0 51        LD   D,C
15D1 59        LD   E,C
15D2 CDA208    CALL 08A2H
15D5 211007    LD   HL,07m0H
15D8 E5        PUSH HL
15D9 21E315    LD   HL,15E3H
15DC CD9A14    CALL 149AH
15DF 218B15    LD   HL,158BH
15E2
15E3 09        ADD  HL,BC
15E4 4A        LD   C,D
15E5 D7        RST  10H
15E6 3B        DEC  SP
15E7 78        LD   A,B
15E8 02        LD   (BC),A
15E9 6E        LD   L,(HL)
15EA 84        ADD  A,H
15EB 7B        LD   A,E
15EC FEC1      CP   0C1H
15EE 2F        CPL
15EF 7C        LD   A,H
15F0 74        LD   (HL),H
15F1 319A7D    LD   SP,7D9AH
15F4 84        ADD  A,H
15F5 3D        DEC  A
15F6 5A        LD   E,D
15F7 7D        LD   A,L
15F8
15F9 7F        LD   A,A
15FA 91        SUB  C
15FB 7E        LD   A,(HL)
15FC E4BB4C    CALL PO,4CBBH
15FF 7E        LD   A,(HL)
1600 6C        LD   L,H
1601 AA        XOR  D
1602 AA        XOR  D
1603 7F        LD   A,A
1604 00        NOP
1605 00        NOP
1606 00        NOP
1607 81        ADD  A,C
1608 8A        ADC  A,D
1609 09        ADD  HL,BC
160A 37        SCF
160B 0B        DEC  BC
```

```
160C 77        LD   (HL),A
160D 09        ADD  HL,BC
160E D427EF    CALL NC,0EF27H
1611 2AF527    LD   HL,(27F5H)
1614 E7        RST  20H
1615 13        INC  DE
1616
1617 14        INC  D
1618 09        ADD  HL,BC
1619 08        EX   AF,AF'
161A 39        ADD  HL,SP
161B 14        INC  D
161C 41        LD   B,C
161D 15        DEC  D
161E 47        LD   B,A
161F 15        DEC  D
1620 A8        XOR  B
1621 15        DEC  D
1622 BD        CP   L
1623
1624 AA        XOR  D
1625 2C        INC  L
1626 52        LD   D,D
1627 41        LD   B,C
1628 58        LD   E,B
1629 41        LD   B,C
162A 5E        LD   E,(HL)
162B 41        LD   B,C
162C 61        LD   H,C
162D 41        LD   B,C
162E 64        LD   H,H
162F 41        LD   B,C
1630 67        LD   H,A
1631 41        LD   B,C
1632 6A        LD   L,D
1633 41        LD   B,C
1634 6D        LD   L,L
1635 41        LD   B,C
1636 70        LD   (HL),B
1637 41        LD   B,C
1638 7F        LD   A,A
1639 0A        LD   A,(BC)
163A B1        OR   C
163B 0A        LD   A,(BC)
163C DB0A      IN   A,(0AH)
163E 260B      LD   H,0BH
1640 03        INC  BC
1641 2A3628    LD   HL,(2836H)
1644 C5        PUSH BC
1645 2A0F2A    LD   HL,(2A0FH)
1648 1F        RRA
1649 2A612A    LD   HL,(2A61H)
164C 91        SUB  C
164D 2A9A2A    LD   HL,(2A9AH)
1650 C5        PUSH BC
1651 4E        LD   C,(HL)
1652 44        LD   B,H
1653 C64F      ADD  4FH
1655 52        LD   D,D
1656 D24553    JP   NC,5345H
1659 45        LD   B,L
165A 54        LD   D,H
165B D345      OUT  (45H),A
165D 54        LD   D,H
165E C34C53    JP   534CH
1661 C34D44    JP   444DH
1664 D2414E    JP   NC,4E41H
1667 44        LD   B,H
1668 4F        LD   C,A
1669 4D        LD   C,L
166A CE45      ADC  45H
166C 58        LD   E,B
166D 54        LD   D,H
166E C44154    CALL NZ,5441H
1671 41        LD   B,C
1672
1673 4E        LD   C,(HL)
1674 50        LD   D,B
1675 55        LD   D,L
1676 54        LD   D,H
1677 C4494D    CALL NZ,4D49H
167A D24541    JP   NC,4145H
167D 44        LD   B,H
167E CC4554    CALL Z,5445H
1681 C7        RST  00H
1682 4F        LD   C,A
1683 54        LD   D,H
1684 4F        LD   C,A
1685 D2554E    JP   NC,4E55H
1688
1689 46        LD   B,(HL)
168A D24553    JP   NC,5345H
168D 54        LD   D,H
168E 4F        LD   C,A
168F 52        LD   D,D
1690 45        LD   B,L
1691 C7        RST  00H
1692 4F        LD   C,A
1693 53        LD   D,E
1694 55        LD   D,L
1695 42        LD   B,D
1696 D24554    JP   NC,5445H
1699 55        LD   D,L
169A 52        LD   D,D
169B 4E        LD   C,(HL)
169C D2454D    JP   NC,4D45H
169F D354      OUT  (54H),A
16A1 4F        LD   C,A
16A2 50        LD   D,B
16A3 C5        PUSH BC
16A4 4C        LD   C,H
16A5 53        LD   D,E
16A6 45        LD   B,L
16A7 D4524F    CALL NC,4F52H
16AA 4E        LD   C,(HL)
16AB D4524F    CALL NC,4F52H
16AE 46        LD   B,(HL)
16AF 46        LD   B,(HL)
16B0 C44546    CALL NZ,4645H
16B3 53        LD   D,E
```

```
16B4 54          LD    D,H
16B5 52          LD    D,D
16B6 C44546      CALL  NZ,4645H
16B9 49          LD    C,C
16BA 4E          LD    C,(HL)
16BB 54          LD    D,H
16BC C44546      CALL  NZ,4645H
16BF 53          LD    D,E
16C0 4E          LD    C,(HL)
16C1 47          LD    B,A
16C2 C44546      CALL  NZ,4645H
16C5 44          LD    B,H
16C6 42          LD    B,D
16C7 4C          LD    C,H
16C8 CC494E      CALL  Z,4E49H
16CB 45          LD    B,L
16CC
16CD 44          LD    B,H
16CE 49          LD    C,C
16CF 54          LD    D,H
16D0
16D1 52          LD    D,D
16D2 52          LD    D,D
16D3 4F          LD    C,A
16D4 52          LD    D,D
16D5 D24553      JP    NC,5345H
16D8 55          LD    D,L
16D9 4D          LD    C,L
16DA 45          LD    B,L
16DB CF          RST   08H
16DC 55          LD    D,L
16DD 54          LD    D,H
16DE CF          RST   08H
16DF 4E          LD    C,(HL)
16E0 CF          RST   08H
16E1 50          LD    D,B
16E2 45          LD    B,L
16E3 4E          LD    C,(HL)
16E4 C649        ADD   49H
16E6 45          LD    B,L
16E7 4C          LD    C,H
16E8 44          LD    B,H
16E9 C7          RST   00H
16EA 45          LD    B,L
16EB 54          LD    D,H
16EC D0          RET   NC
16ED 55          LD    D,L
16EE 54          LD    D,H
16EF C34C4F      JP    4F4CH
16F2 53          LD    D,E
16F3 45          LD    B,L
16F4 CC4F41      CALL  Z,414FH
16F7 44          LD    B,H
16F8 CD4552      CALL  5245H
16FB 47          LD    B,A
16FC 45          LD    B,L
16FD CE41        ADC   41H
16FF 4D          LD    C,L
1700 45          LD    B,L
1701 CB49        BIT   01H,C
1703 4C          LD    C,H
1704 4C          LD    C,H
1705 CC5345      CALL  Z,4553H
1708 54          LD    D,H
1709 D25345      JP    NC,4553H
170C 54          LD    D,H
170D D341        OUT   (41H),A
170F 56          LD    D,(HL)
1710 45          LD    B,L
1711 D359        OUT   (59H),A
1713 53          LD    D,E
1714 54          LD    D,H
1715 45          LD    B,L
1716 4D          LD    C,L
1717 CC5052      CALL  Z,5250H
171A 49          LD    C,C
171B 4E          LD    C,(HL)
171C 54          LD    D,H
171D C44546      CALL  NZ,4645H
1720
1721 4F          LD    C,A
1722 4B          LD    C,E
1723 45          LD    B,L
1724
1725 52          LD    D,D
1726 49          LD    C,C
1727 4E          LD    C,(HL)
1728 54          LD    D,H
1729 C34F4E      JP    4E4FH
172C 54          LD    D,H
172D CC4953      CALL  Z,5349H
1730 54          LD    D,H
1731 CC4C49      CALL  Z,494CH
1734 53          LD    D,E
1735 54          LD    D,H
1736 C44454C     CALL  NZ,4C45H
1739 45          LD    B,L
173A 54          LD    D,H
173B 45          LD    B,L
173C C1          POP   BC
173D 55          LD    D,L
173E 54          LD    D,H
173F 4F          LD    C,A
1740 C34C45      JP    454CH
1743 41          LD    B,C
1744 52          LD    D,D
1745 C34C4F      JP    4F4CH
1748 41          LD    B,C
1749 44          LD    B,H
174A C3534E1     JP    E1534H
174D 56          LD    D,(HL)
174E 45          LD    B,L
174F CE45        ADC   45H
17451 57         LD    D,A
17452 D4441442   CALL  NC,4214H
17455 28D4       JR    Z,1722BH
17457 4F         LD    C,A
17458 C64E       ADD   4EH
1745A D5         PUSH  DE
1745B 53         LD    D,E
```

```
175C 49        LD   C,C
175D 4E        LD   C,(HL)
175E 47        LD   B,A
175F D641      SUB  41H
1761 52        LD   D,D
1762 50        LD   D,B
1763 54        LD   D,H
1764 52        LD   D,D
1765 D5        PUSH DE
1766 53        LD   D,E
1767 52        LD   D,D
1768 C5        PUSH BC
1769 52        LD   D,D
176A 4C        LD   C,H
176B C5        PUSH BC
176C 52        LD   D,D
176D 52        LD   D,D
176E D354      OUT  (54H),A
1770 52        LD   D,D
1771 49        LD   C,C
1772 4E        LD   C,(HL)
1773 47        LD   B,A
1774 24        INC  H
1775
1776 4E        LD   C,(HL)
1777 53        LD   D,E
1778 54        LD   D,H
1779 52        LD   D,D
177A D0        RET  NC
177B 4F        LD   C,A
177C 49        LD   C,C
177D 4E        LD   C,(HL)
177E 54        LD   D,H
177F D4494D    CALL NC,4D49H
1782 45        LD   B,L
1783 24        INC  H
1784 CD454D    CALL 4D45H
1787
1788 4E        LD   C,(HL)
1789 4B        LD   C,E
178A 45        LD   B,L
178B 59        LD   E,C
178C 24        INC  H
178D D44845    CALL NC,4548H
1790 4E        LD   C,(HL)
1791 CE4F      ADC  4FH
1793 54        LD   D,H
1794 D354      OUT  (54H),A
1796 45        LD   B,L
1797 50        LD   D,B
1798 AB        XOR  E
1799 AD        XOR  L
179A AA        XOR  D
179B AF        XOR  A
179C DBC1      IN   A,(0C1H)
179E 4E        LD   C,(HL)
179F 44        LD   B,H
17A0 CF        RST  08H
17A1 52        LD   D,D
17A2 BE        CP   (HL)
17A3 BD        CP   L
17A4 BC        CP   H
17A5 D347      OUT  (47H),A
17A7 4E        LD   C,(HL)
17A8
17A9 4E        LD   C,(HL)
17AA 54        LD   D,H
17AB C1        POP  BC
17AC 42        LD   B,D
17AD 53        LD   D,E
17AE C652      ADD  52H
17B0 45        LD   B,L
17B1
17B2 4E        LD   C,(HL)
17B3 50        LD   D,B
17B4 D0        RET  NC
17B5 4F        LD   C,A
17B6 53        LD   D,E
17B7 D351      OUT  (51H),A
17B9 52        LD   D,D
17BA D24E44    JP   NC,444EH
17BD CC4F47    CALL Z,474FH
17C0 C5        PUSH BC
17C1 58        LD   E,B
17C2 50        LD   D,B
17C3 C34F53    JP   534FH
17C6 D349      OUT  (49H),A
17C8 4E        LD   C,(HL)
17C9 D4414E    CALL NC,4E41H
17CC C1        POP  BC
17CD 54        LD   D,H
17CE 4E        LD   C,(HL)
17CF D0        RET  NC
17D0 45        LD   B,L
17D1 45        LD   B,L
17D2 4B        LD   C,E
17D3 C35649    JP   4956H
17D6 C35653    JP   5356H
17D9 C35644    JP   4456H
17DC C5        PUSH BC
17DD 4F        LD   C,A
17DE 46        LD   B,(HL)
17DF CC4F43    CALL Z,434FH
17E2 CC4F46    CALL Z,464FH
17E5 CD4B49    CALL 494BH
17E8 24        INC  H
17E9 CD4B53    CALL 534BH
17EC 24        INC  H
17ED CD4B44    CALL 444BH
17F0 24        INC  H
17F1 C3494E    JP   4E49H
17F4 54        LD   D,H
17F5 C3534E    JP   4E53H
17F8 47        LD   B,A
17F9 C34442    JP   4244H
17FC 4C        LD   C,H
17FD C649      ADD  49H
17FF 58        LD   E,B
1800 CC454E    CALL Z,4E45H
1803 D354      OUT  (54H),A
```

```
1805 52        LD   D,D
1806 24        INC  H
1807 D641      SUB  41H
1809 4C        LD   C,H
180A C1        POP  BC
180B 53        LD   D,E
180C 43        LD   B,E
180D C34852    JP   5248H
1810 24        INC  H
1811 CC4546    CALL Z,4645H
1814 54        LD   D,H
1815 24        INC  H
1816 D24947    JP   NC,4749H
1819 48        LD   C,B
181A 54        LD   D,H
181B 24        INC  H
181C CD4944    CALL 4449H
181F 24        INC  H
1820 A7        AND  A
1821 80        ADD  A,B
1822 AE        XOR  (HL)
1823 1D        DEC  E
1824 A1        AND  C
1825 1C        INC  E
1826 3801      JR   C,1829H
1828 35        DEC  (HL)
1829 01C901    LD   BC,01C9H
182C 73        LD   (HL),E
182D 41        LD   B,C
182E D301      OUT  (01H),A
1830 B6        OR   (HL)
1831 22051F    LD   (1F05H),HL
1834 9A        SBC  A,D
1835 210826    LD   HL,2608H
1838 EF        RST  28H
1839 21211F    LD   HL,1F21H
183C C21EA3    JP   NZ,0A31EH
183F 1E39      LD   E,39H
1841 2091      JR   NZ,17D4H
1843
1844 B1        OR   C
1845 1EDE      LD   E,0DEH
1847 1E07      LD   E,07H
1849 1F        RRA
184A A9        XOR  C
184B 1D        DEC  E
184C 07        RLCA
184D 1F        RRA
184E F7        RST  30H
184F 1D        DEC  E
1850 F8        RET  M
1851
1852 00        NOP
1853 1E03      LD   E,03H
1855 1E06      LD   E,06H
1857 1E09      LD   E,09H
1859 1EA3      LD   E,0A3H
185B 41        LD   B,C
185C 60        LD   H,B
185D 2EF4      LD   L,0F4H
185F 1F        RRA
1860 AF        XOR  A
1861 1F        RRA
1862 FB        EI
1863 2A6C1F    LD   HL,(1F6CH)
1866 79        LD   A,C
1867 41        LD   B,C
1868 7C        LD   A,H
1869 41        LD   B,C
186A 7F        LD   A,A
186B 41        LD   B,C
186C 82        ADD  A,D
186D 41        LD   B,C
186E 85        ADD  A,L
186F 41        LD   B,C
1870 88        ADC  A,B
1871 41        LD   B,C
1872 8B        ADC  A,E
1873 41        LD   B,C
1874 8E        ADC  A,(HL)
1875 41        LD   B,C
1876 91        SUB  C
1877 41        LD   B,C
1878 97        SUB  A
1879 41        LD   B,C
187A 9A        SBC  A,D
187B 41        LD   B,C
187C A0        AND  B
187D 41        LD   B,C
187E B2        OR   D
187F 02        LD   (BC),A
1880 67        LD   H,A
1881 205B      JR   NZ,18DEH
1883 41        LD   B,C
1884 B1        OR   C
1885 2C        INC  L
1886 6F        LD   L,A
1887 20E4      JR   NZ,186DH
1889 1D        DEC  E
188A 2E2B      LD   L,2BH
188C 29        ADD  HL,HL
188D 2B        DEC  HL
188E C62B      ADD  2BH
1890 08        EX   AF,AF'
1891 207A      JR   NZ,190DH
1893 1E1F      LD   E,1FH
1895 2C        INC  L
1896 F5        PUSH AF
1897 2B        DEC  HL
1898 49        LD   C,C
1899 1B        DEC  DE
189A 79        LD   A,C
189B 79        LD   A,C
189C
189D
189E 7F        LD   A,A
189F 50        LD   D,B
18A0 46        LD   B,(HL)
18A1 DB0A      IN   A,(0AH)
18A3 00        NOP
```

```
18A4 00        NOP
18A5 7F        LD    A,A
18A6 0A        LD    A,(BC)
18A7 F40AB1    CALL  P,0B10AH
18AA 0A        LD    A,(BC)
18AB 77        LD    (HL),A
18AC
18AD 70        LD    (HL),B
18AE
18AF A1        AND   C
18B0 0D        DEC   C
18B1 E5        PUSH  HL
18B2 0D        DEC   C
18B3 78        LD    A,B
18B4 0A        LD    A,(BC)
18B5 1607      LD    D,07H
18B7 13        INC   DE
18B8 07        RLCA
18B9 47        LD    B,A
18BA 08        EX    AF,AF'
18BB A2        AND   D
18BC 08        EX    AF,AF'
18BD 0C        INC   C
18BE 0A        LD    A,(BC)
18BF D20BC7    JP    NC,0C70BH
18C2 0B        DEC   BC
18C3 F20B90    JP    P,900BH
18C6 24        INC   H
18C7 39        ADD   HL,SP
18C8 0A        LD    A,(BC)
18C9 4E        LD    C,(HL)
18CA 46        LD    B,(HL)
18CB 53        LD    D,E
18CC 4E        LD    C,(HL)
18CD 52        LD    D,D
18CE 47        LD    B,A
18CF 4F        LD    C,A
18D0 44        LD    B,H
18D1 46        LD    B,(HL)
18D2 43        LD    B,E
18D3 4F        LD    C,A
18D4 56        LD    D,(HL)
18D5 4F        LD    C,A
18D6 4D        LD    C,L
18D7 55        LD    D,L
18D8 4C        LD    C,H
18D9 42        LD    B,D
18DA 53        LD    D,E
18DB 44        LD    B,H
18DC 44        LD    B,H
18DD 2F        CPL
18DE 3049      JR    NC,1929H
18E0 44        LD    B,H
18E1 54        LD    D,H
18E2 4D        LD    C,L
18E3 4F        LD    C,A
18E4 53        LD    D,E
18E5 4C        LD    C,H
18E6 53        LD    D,E
18E7 53        LD    D,E
18E8 54        LD    D,H
18E9 43        LD    B,E
18EA 4E        LD    C,(HL)
18EB 4E        LD    C,(HL)
18EC 52        LD    D,D
18ED 52        LD    D,D
18EE 57        LD    D,A
18EF 55        LD    D,L
18F0 45        LD    B,L
18F1 4D        LD    C,L
18F2 4F        LD    C,A
18F3 46        LD    B,(HL)
18F4 44        LD    B,H
18F5 4C        LD    C,H
18F6 33        INC   SP
18F7 D600      SUB   00H
18F9 6F        LD    L,A
18FA 7C        LD    A,H
18FB DE00      SBC   00H
18FD 67        LD    H,A
18FE 78        LD    A,B
18FF DE00      SBC   00H
1901 47        LD    B,A
1902 3E00      LD    A,00H
1904 C9        RET
1905 4A        LD    C,D
1906 1E40      LD    E,40H
1908 E64D      AND   4DH
190A DB00      IN    A,(00H)
190C
190D D300      OUT   (00H),A
190F
1910 00        NOP
1911 00        NOP
1912 00        NOP
1913 00        NOP
1914 40        LD    B,B
1915 3000      JR    NC,1917H
1917 4C        LD    C,H
1918 43        LD    B,E
1919 FEFF      CP    0FFH
191B E9        JP    (HL)
191C 42        LD    B,D
191D 2045      JR    NZ,1964H
191F 72        LD    (HL),D
1920 72        LD    (HL),D
1921 6F        LD    L,A
1922 72        LD    (HL),D
1923 00        NOP
1924 2069      JR    NZ,198FH
1926 6E        LD    L,(HL)
1927 2000      JR    NZ,1929H
1929 52        LD    D,D
192A 45        LD    B,L
192B 41        LD    B,C
192C 44        LD    B,H
192D 59        LD    E,C
192E 0D        DEC   C
192F 00        NOP
1930 42        LD    B,D
```

```
1931 72        LD   (HL),D
1932 65        LD   H,L
1933 61        LD   H,C
1934 6B        LD   L,E
1935 00        NOP
1936 210400    LD   HL,0004H
1939 39        ADD  HL,SP
193A 7E        LD   A,(HL)
193B 23        INC  HL
193C FE81      CP   81H
193E C0        RET  NZ
193F 4E        LD   C,(HL)
1940 23        INC  HL
1941 46        LD   B,(HL)
1942 23        INC  HL
1943 E5        PUSH HL
1944 69        LD   L,C
1945 60        LD   H,B
1946 7A        LD   A,D
1947 B3        OR   E
1948 EB        EX   DE,HL
1949 2802      JR   Z,194DH
194B EB        EX   DE,HL
194C
194D 010E00    LD   BC,000EH
1950 E1        POP  HL
1951 C8        RET  Z
1952 09        ADD  HL,BC
1953 18E5      JR   193AH
1955 CD6C19    CALL 196CH
1958 C5        PUSH BC
1959 E3        EX   (SP),HL
195A C1        POP  BC
195B
195C 7E        LD   A,(HL)
195D 02        LD   (BC),A
195E C8        RET  Z
195F 0B        DEC  BC
1960 2B        DEC  HL
1961 18F8      JR   195BH
1963 E5        PUSH HL
1964 2AFD40    LD   HL,(40FDH)
1967 0600      LD   B,00H
1969 09        ADD  HL,BC
196A 09        ADD  HL,BC
196B 3EE5      LD   A,0E5H
196D 3EC6      LD   A,0C6H
196F 95        SUB  L
1970 6F        LD   L,A
1971 3EFF      LD   A,0FFH
1973 9C        SBC  A,H
1974 3804      JR   C,197AH
1976 67        LD   H,A
1977 39        ADD  HL,SP
1978 E1        POP  HL
1979 D8        RET  C
197A 1E0C      LD   E,0CH
197C 1824      JR   19A2H
197E 2AA240    LD   HL,(40A2H)
1981 7C        LD   A,H
1982 A5        AND  L
1983
1984 2808      JR   Z,198EH
1986 3AF240    LD   A,(40F2H)
1989 B7        OR   A
198A 1E22      LD   E,22H
198C 2014      JR   NZ,19A2H
198E C3C11D    JP   1DC1H
1991 2ADA40    LD   HL,(40DAH)
1994 22A240    LD   (40A2H),HL
1997 1E02      LD   E,02H
1999 011E14    LD   BC,141EH
199C 011E00    LD   BC,001EH
199F 011E24    LD   BC,241EH
19A2 2AA240    LD   HL,(40A2H)
19A5 22EA40    LD   (40EAH),HL
19A8 22EC40    LD   (40ECH),HL
19AB 01B419    LD   BC,19B4H
19AE 2AE840    LD   HL,(40E8H)
19B1 C39A1B    JP   1B9AH
19B4 C1        POP  BC
19B5 7B        LD   A,E
19B6 4B        LD   C,E
19B7 329A40    LD   (409AH),A
19BA 2AE640    LD   HL,(40E6H)
19BD 22EE40    LD   (40EEH),HL
19C0 EB        EX   DE,HL
19C1 2AEA40    LD   HL,(40EAH)
19C4 7C        LD   A,H
19C5 A5        AND  L
19C6
19C7 2807      JR   Z,19D0H
19C9 22F540    LD   (40F5H),HL
19CC EB        EX   DE,HL
19CD 22F740    LD   (40F7H),HL
19D0 2AF040    LD   HL,(40F0H)
19D3 7C        LD   A,H
19D4 B5        OR   L
19D5 EB        EX   DE,HL
19D6 21F240    LD   HL,40F2H
19D9 2808      JR   Z,19E3H
19DB A6        AND  (HL)
19DC 2005      JR   NZ,19E3H
19DE 35        DEC  (HL)
19DF EB        EX   DE,HL
19E0 C3361D    JP   1D36H
19E3 AF        XOR  A
19E4 77        LD   (HL),A
19E5 59        LD   E,C
19E6 CDF920    CALL 20F9H
19E9 21C918    LD   HL,18C9H
19EC CDA641    CALL 41A6H
19EF 57        LD   D,A
19F0 3E3F      LD   A,3FH
19F2 CD2A03    CALL 032AH
19F5 19        ADD  HL,DE
19F6 7E        LD   A,(HL)
19F7 CD2A03    CALL 032AH
19FA D7        RST  10H
19FB CD2A03    CALL 032AH
```

```
19FE 211D19    LD   HL,191DH
1A01 E5        PUSH HL
1A02 2AEA40    LD   HL,(40EAH)
1A05 E3        EX   (SP),HL
1A06 CDA728    CALL 28A7H
1A09 E1        POP  HL
1A0A 11FEFF    LD   DE,0FFFEH
1A0D
1A0E CA7406    JP   Z,0674H
1A11 7C        LD   A,H
1A12 A5        AND  L
1A13 3C        INC  A
1A14 C4A70F    CALL NZ,0FA7H
1A17 3EC1      LD   A,0C1H
1A19 CD8B03    CALL 038BH
1A1C CDAC41    CALL 41ACH
1A1F CDF801    CALL 01F8H
1A22 CDF920    CALL 20F9H
1A25 212919    LD   HL,1929H
1A28 CDA728    CALL 28A7H
1A2B 3A9A40    LD   A,(409AH)
1A2E D602      SUB  02H
1A30 CC532E    CALL Z,2E53H
1A33 21FFFF    LD   HL,0FFFFH
1A36 22A240    LD   (40A2H),HL
1A39 3AE140    LD   A,(40E1H)
1A3C B7        OR   A
1A3D 2837      JR   Z,1A76H
1A3F 2AE240    LD   HL,(40E2H)
1A42 E5        PUSH HL
1A43 CDAF0F    CALL 0FAFH
1A46 D1        POP  DE
1A47 D5        PUSH DE
1A48 CD2C1B    CALL 1B2CH
1A4B 3E2A      LD   A,2AH
1A4D 3802      JR   C,1A51H
1A4F 3E20      LD   A,20H
1A51 CD2A03    CALL 032AH
1A54 CD6103    CALL 0361H
1A57 D1        POP  DE
1A58 3006      JR   NC,1A60H
1A5A AF        XOR  A
1A5B 32E140    LD   (40E1H),A
1A5E 18B9      JR   1A19H
1A60 2AE440    LD   HL,(40E4H)
1A63 19        ADD  HL,DE
1A64 38F4      JR   C,1A5AH
1A66 D5        PUSH DE
1A67 11F9FF    LD   DE,0FFF9H
1A6A
1A6B D1        POP  DE
1A6C 30EC      JR   NC,1A5AH
1A6E 22E240    LD   (40E2H),HL
1A71 F6FF      OR   0FFH
1A73 C3EB2F    JP   2FEBH
1A76 3E3E      LD   A,3EH
1A78 CD2A03    CALL 032AH
1A7B CD6103    CALL 0361H
1A7E DA331A    JP   C,1A33H
1A81 D7        RST  10H
1A82 3C        INC  A
1A83 3D        DEC  A
1A84 CA331A    JP   Z,1A33H
1A87 F5        PUSH AF
1A88 CD5A1E    CALL 1E5AH
1A8B 2B        DEC  HL
1A8C 7E        LD   A,(HL)
1A8D FE20      CP   20H
1A8F 28FA      JR   Z,1A8BH
1A91 23        INC  HL
1A92 7E        LD   A,(HL)
1A93 FE20      CP   20H
1A95 CCC909    CALL Z,09C9H
1A98 D5        PUSH DE
1A99 CDC01B    CALL 1BC0H
1A9C D1        POP  DE
1A9D F1        POP  AF
1A9E 22E640    LD   (40E6H),HL
1AA1 CDB241    CALL 41B2H
1AA4 D25A1D    JP   NC,1D5AH
1AA7 D5        PUSH DE
1AA8 C5        PUSH BC
1AA9 AF        XOR  A
1AAA 32DD40    LD   (40DDH),A
1AAD D7        RST  10H
1AAE B7        OR   A
1AAF F5        PUSH AF
1AB0 EB        EX   DE,HL
1AB1 22EC40    LD   (40ECH),HL
1AB4
1AB5 CD2C1B    CALL 1B2CH
1AB8 C5        PUSH BC
1AB9 DCE42B    CALL C,2BE4H
1ABC D1        POP  DE
1ABD F1        POP  AF
1ABE D5        PUSH DE
1ABF 2827      JR   Z,1AE8H
1AC1 D1        POP  DE
1AC2 2AF940    LD   HL,(40F9H)
1AC5 E3        EX   (SP),HL
1AC6 C1        POP  BC
1AC7 09        ADD  HL,BC
1AC8 E5        PUSH HL
1AC9 CD5519    CALL 1955H
1ACC E1        POP  HL
1ACD 22F940    LD   (40F9H),HL
1AD0
1AD1 74        LD   (HL),H
1AD2 D1        POP  DE
1AD3 E5        PUSH HL
1AD4 23        INC  HL
1AD5 23        INC  HL
1AD6 73        LD   (HL),E
1AD7 23        INC  HL
1AD8 72        LD   (HL),D
1AD9 23        INC  HL
1ADA EB        EX   DE,HL
1ADB 2AA740    LD   HL,(40A7H)
1ADE EB        EX   DE,HL
1ADF 1B        DEC  DE
```

```
1AE0 1B       DEC  DE
1AE1 1A       LD   A,(DE)
1AE2 77       LD   (HL),A
1AE3 23       INC  HL
1AE4 13       INC  DE
1AE5 B7       OR   A
1AE6 20F9     JR   NZ,1AE1H
1AE8 D1       POP  DE
1AE9 CDFC1A   CALL 1AFCH
1AEC CDB541   CALL 41B5H
1AEF CD5D1B   CALL 1B5DH
1AF2 CDB841   CALL 41B8H
1AF5 C3331A   JP   1A33H
1AF8 2AA440   LD   HL,(40A4H)
1AFB EB       EX   DE,HL
1AFC 62       LD   H,D
1AFD 6B       LD   L,E
1AFE 7E       LD   A,(HL)
1AFF
1B00 B6       OR   (HL)
1B01 C8       RET  Z
1B02 23       INC  HL
1B03 23       INC  HL
1B04 23       INC  HL
1B05 AF       XOR  A
1B06 BE       CP   (HL)
1B07 23       INC  HL
1B08 20FC     JR   NZ,1B06H
1B0A EB       EX   DE,HL
1B0B 73       LD   (HL),E
1B0C
1B0D 72       LD   (HL),D
1B0E 18EC     JR   1AFCH
1B10 110000   LD   DE,0000H
1B13 D5       PUSH DE
1B14 2809     JR   Z,1B1FH
1B16 D1       POP  DE
1B17 CD4F1E   CALL 1E4FH
1B1A D5       PUSH DE
1B1B 280B     JR   Z,1B28H
1B1D CF       RST  08H
1B1E CE11     ADC  11H
1B20 FAFFC4   JP   M,0C4FFH
1B23 4F       LD   C,A
1B24 1EC2     LD   E,0C2H
1B26 97       SUB  A
1B27 19       ADD  HL,DE
1B28 EB       EX   DE,HL
1B29 D1       POP  DE
1B2A E3       EX   (SP),HL
1B2B E5       PUSH HL
1B2C 2AA440   LD   HL,(40A4H)
1B2F 44       LD   B,H
1B30 4D       LD   C,L
1B31 7E       LD   A,(HL)
1B32 23       INC  HL
1B33 B6       OR   (HL)
1B34 2B       DEC  HL
1B35 C8       RET  Z
1B36 23       INC  HL
1B37 23       INC  HL
1B38 7E       LD   A,(HL)
1B39 23       INC  HL
1B3A 66       LD   H,(HL)
1B3B 6F       LD   L,A
1B3C
1B3D 60       LD   H,B
1B3E 69       LD   L,C
1B3F 7E       LD   A,(HL)
1B40 23       INC  HL
1B41 66       LD   H,(HL)
1B42 6F       LD   L,A
1B43 3F       CCF
1B44 C8       RET  Z
1B45 3F       CCF
1B46 D0       RET  NC
1B47 18E6     JR   1B2FH
1B49 C0       RET  NZ
1B4A CDC901   CALL 01C9H
1B4D 2AA440   LD   HL,(40A4H)
1B50 CDF81D   CALL 1DF8H
1B53 32E140   LD   (40E1H),A
1B56 77       LD   (HL),A
1B57 23       INC  HL
1B58 77       LD   (HL),A
1B59 23       INC  HL
1B5A 22F940   LD   (40F9H),HL
1B5D 2AA440   LD   HL,(40A4H)
1B60 2B       DEC  HL
1B61 22DF40   LD   (40DFH),HL
1B64 061A     LD   B,1AH
1B66 210141   LD   HL,4101H
1B69 3604     LD   (HL),04H
1B6B 23       INC  HL
1B6C 10FB     DJNZ 1B69H
1B6E AF       XOR  A
1B6F 32F240   LD   (40F2H),A
1B72 6F       LD   L,A
1B73 67       LD   H,A
1B74 22F040   LD   (40F0H),HL
1B77 22F740   LD   (40F7H),HL
1B7A 2AB140   LD   HL,(40B1H)
1B7D 22D640   LD   (40D6H),HL
1B80 CD911D   CALL 1D91H
1B83 2AF940   LD   HL,(40F9H)
1B86 22FB40   LD   (40FBH),HL
1B89 22FD40   LD   (40FDH),HL
1B8C CDBB41   CALL 41BBH
1B8F C1       POP  BC
1B90 2AA040   LD   HL,(40A0H)
1B93 2B       DEC  HL
1B94 2B       DEC  HL
1B95 22E840   LD   (40E8H),HL
1B98
1B99 23       INC  HL
1B9A F9       LD   SP,HL
1B9B 21B540   LD   HL,40B5H
1B9E 22B340   LD   (40B3H),HL
1BA1 CD8B03   CALL 038BH
1BA4 CD6921   CALL 2169H
```

```
1BA7 AF        XOR  A
1BA8 67        LD   H,A
1BA9 6F        LD   L,A
1BAA 32DC40    LD   (40DCH),A
1BAD E5        PUSH HL
1BAE C5        PUSH BC
1BAF 2ADF40    LD   HL,(40DFH)
1BB2 C9        RET
1BB3 3E3F      LD   A,3FH
1BB5 CD2A03    CALL 032AH
1BB8 3E20      LD   A,20H
1BBA CD2A03    CALL 032AH
1BBD C36103    JP   0361H
1BC0 AF        XOR  A
1BC1 32B040    LD   (40B0H),A
1BC4 4F        LD   C,A
1BC5 EB        EX   DE,HL
1BC6 2AA740    LD   HL,(40A7H)
1BC9 2B        DEC  HL
1BCA 2B        DEC  HL
1BCB EB        EX   DE,HL
1BCC 7E        LD   A,(HL)
1BCD FE20      CP   20H
1BCF CA5B1C    JP   Z,1C5BH
1BD2 47        LD   B,A
1BD3 FE22      CP   22H
1BD5 CA771C    JP   Z,1C77H
1BD8 B7        OR   A
1BD9 CA7D1C    JP   Z,1C7DH
1BDC 3AB040    LD   A,(40B0H)
1BDF B7        OR   A
1BE0 7E        LD   A,(HL)
1BE1 C25B1C    JP   NZ,1C5BH
1BE4 FE3F      CP   3FH
1BE6 3EB2      LD   A,0B2H
1BE8 CA5B1C    JP   Z,1C5BH
1BEB 7E        LD   A,(HL)
1BEC FE30      CP   30H
1BEE 3805      JR   C,1BF5H
1BF0 FE3C      CP   3CH
1BF2 DA5B1C    JP   C,1C5BH
1BF5 D5        PUSH DE
1BF6 114F16    LD   DE,164FH
1BF9
1BFA 013D1C    LD   BC,1C3DH
1BFD
1BFE 067F      LD   B,7FH
1C00 7E        LD   A,(HL)
1C01 FE61      CP   61H
1C03 3807      JR   C,1C0CH
1C05 FE7B      CP   7BH
1C07 3003      JR   NC,1C0CH
1C09 E65F      AND  5FH
1C0B 77        LD   (HL),A
1C0C 4E        LD   C,(HL)
1C0D EB        EX   DE,HL
1C0E 23        INC  HL
1C0F B6        OR   (HL)
1C10 F20E1C    JP   P,1C0EH
1C13 04        INC  B
1C14 7E        LD   A,(HL)
1C15 E67F      AND  7FH
1C17 C8        RET  Z
1C18 B9        CP   C
1C19 20F3      JR   NZ,1C0EH
1C1B EB        EX   DE,HL
1C1C E5        PUSH HL
1C1D 13        INC  DE
1C1E 1A        LD   A,(DE)
1C1F B7        OR   A
1C20 FA391C    JP   M,1C39H
1C23 4F        LD   C,A
1C24 78        LD   A,B
1C25 FE8D      CP   8DH
1C27 2002      JR   NZ,1C2BH
1C29 D7        RST  10H
1C2A 2B        DEC  HL
1C2B 23        INC  HL
1C2C 7E        LD   A,(HL)
1C2D FE61      CP   61H
1C2F 3802      JR   C,1C33H
1C31 E65F      AND  5FH
1C33 B9        CP   C
1C34 28E7      JR   Z,1C1DH
1C36 E1        POP  HL
1C37 18D3      JR   1C0CH
1C39 48        LD   C,B
1C3A F1        POP  AF
1C3B
1C3C C9        RET
1C3D EB        EX   DE,HL
1C3E 79        LD   A,C
1C3F C1        POP  BC
1C40 D1        POP  DE
1C41 EB        EX   DE,HL
1C42 FE95      CP   95H
1C44 363A      LD   (HL),3AH
1C46 2002      JR   NZ,1C4AH
1C48 0C        INC  C
1C49 23        INC  HL
1C4A FEFB      CP   0FBH
1C4C 200C      JR   NZ,1C5AH
1C4E 363A      LD   (HL),3AH
1C50 23        INC  HL
1C51 0693      LD   B,93H
1C53 70        LD   (HL),B
1C54 23        INC  HL
1C55
1C56 0C        INC  C
1C57 0C        INC  C
1C58 181D      JR   1C77H
1C5A EB        EX   DE,HL
1C5B 23        INC  HL
1C5C 12        LD   (DE),A
1C5D 13        INC  DE
1C5E 0C        INC  C
1C5F D63A      SUB  3AH
1C61 2804      JR   Z,1C67H
1C63 FE4E      CP   4EH
1C65 2003      JR   NZ,1C6AH
```

```
1C67 32B040    LD   (40B0H),A
1C6A D659      SUB  59H
1C6C C2CC1B    JP   NZ,1BCCH
1C6F 47        LD   B,A
1C70 7E        LD   A,(HL)
1C71 B7        OR   A
1C72 2809      JR   Z,1C7DH
1C74 B8        CP   B
1C75 28E4      JR   Z,1C5BH
1C77 23        INC  HL
1C78 12        LD   (DE),A
1C79 0C        INC  C
1C7A 13        INC  DE
1C7B 18F3      JR   1C70H
1C7D 210500    LD   HL,0005H
1C80 44        LD   B,H
1C81 09        ADD  HL,BC
1C82 44        LD   B,H
1C83 4D        LD   C,L
1C84 2AA740    LD   HL,(40A7H)
1C87 2B        DEC  HL
1C88 2B        DEC  HL
1C89 2B        DEC  HL
1C8A 12        LD   (DE),A
1C8B 13        INC  DE
1C8C 12        LD   (DE),A
1C8D 13        INC  DE
1C8E 12        LD   (DE),A
1C8F
1C90 7C        LD   A,H
1C91 92        SUB  D
1C92 C0        RET  NZ
1C93 7D        LD   A,L
1C94 93        SUB  E
1C95
1C96 7E        LD   A,(HL)
1C97 E3        EX   (SP),HL
1C98 BE        CP   (HL)
1C99 23        INC  HL
1C9A E3        EX   (SP),HL
1C9B CA781D    JP   Z,1D78H
1C9E C39719    JP   1997H
1CA1 3E64      LD   A,64H
1CA3 32DC40    LD   (40DCH),A
1CA6 CD211F    CALL 1F21H
1CA9 E3        EX   (SP),HL
1CAA CD3619    CALL 1936H
1CAD D1        POP  DE
1CAE 2005      JR   NZ,1CB5H
1CB0 09        ADD  HL,BC
1CB1 F9        LD   SP,HL
1CB2 22E840    LD   (40E8H),HL
1CB5 EB        EX   DE,HL
1CB6 0E08      LD   C,08H
1CB8 CD6319    CALL 1963H
1CBB E5        PUSH HL
1CBC CD051F    CALL 1F05H
1CBF E3        EX   (SP),HL
1CC0 E5        PUSH HL
1CC1 2AA240    LD   HL,(40A2H)
1CC4 E3        EX   (SP),HL
1CC5 CF        RST  08H
1CC6 BD        CP   L

1CC8 CAF60A    JP   Z,0AF6H
1CCB D2F60A    JP   NC,0AF6H
1CCE F5        PUSH AF
1CCF CD3723    CALL 2337H
1CD2 F1        POP  AF
1CD3 E5        PUSH HL
1CD4 F2EC1C    JP   P,1CECH
1CD7 CD7F0A    CALL 0A7FH
1CDA E3        EX   (SP),HL
1CDB 110100    LD   DE,0001H
1CDE 7E        LD   A,(HL)
1CDF FECC      CP   0CCH
1CE1 CC012B    CALL Z,2B01H
1CE4 D5        PUSH DE
1CE5 E5        PUSH HL
1CE6 EB        EX   DE,HL
1CE7 CD9E09    CALL 099EH
1CEA 1822      JR   1D0EH
1CEC CDB10A    CALL 0AB1H
1CEF CDBF09    CALL 09BFH
1CF2 E1        POP  HL
1CF3 C5        PUSH BC
1CF4 D5        PUSH DE
1CF5 010081    LD   BC,8100H
1CF8 51        LD   D,C
1CF9 5A        LD   E,D
1CFA 7E        LD   A,(HL)
1CFB FECC      CP   0CCH
1CFD 3E01      LD   A,01H
1CFF 200E      JR   NZ,1D0FH
1D01 CD3823    CALL 2338H
1D04 E5        PUSH HL
1D05 CDB10A    CALL 0AB1H
1D08 CDBF09    CALL 09BFH
1D0B CD5509    CALL 0955H
1D0E E1        POP  HL
1D0F C5        PUSH BC
1D10 D5        PUSH DE
1D11 4F        LD   C,A
1D12
1D13 47        LD   B,A
1D14 C5        PUSH BC
1D15 E5        PUSH HL
1D16 2ADF40    LD   HL,(40DFH)
1D19 E3        EX   (SP),HL
1D1A 0681      LD   B,81H
1D1C C5        PUSH BC
1D1D 33        INC  SP
1D1E CD5803    CALL 0358H
1D21 B7        OR   A
1D22 C4A01D    CALL NZ,1DA0H
1D25 22E640    LD   (40E6H),HL
1D28 ED73E840  LD   (40E8H),SP
1D2C 7E        LD   A,(HL)
1D2D FE3A      CP   3AH
1D2F 2829      JR   Z,1D5AH
```

```
1D31 B7        OR   A
1D32 C29719    JP   NZ,1997H
1D35 23        INC  HL
1D36 7E        LD   A,(HL)
1D37 23        INC  HL
1D38 B6        OR   (HL)
1D39 CA7E19    JP   Z,197EH
1D3C 23        INC  HL
1D3D 5E        LD   E,(HL)
1D3E 23        INC  HL
1D3F 56        LD   D,(HL)
1D40 EB        EX   DE,HL
1D41 22A240    LD   (40A2H),HL
1D44 3A1B41    LD   A,(411BH)
1D47 B7        OR   A
1D48 280F      JR   Z,1D59H
1D4A D5        PUSH DE
1D4B 3E3C      LD   A,3CH
1D4D CD2A03    CALL 032AH
1D50 CDAF0F    CALL 0FAFH
1D53 3E3E      LD   A,3EH
1D55 CD2A03    CALL 032AH
1D58 D1        POP  DE
1D59 EB        EX   DE,HL
1D5A D7        RST  10H
1D5B 111E1D    LD   DE,1D1EH
1D5E D5        PUSH DE
1D5F
1D60 D680      SUB  80H
1D62 DA211F    JP   C,1F21H
1D65 FE3C      CP   3CH
1D67 D2E72A    JP   NC,2AE7H
1D6A 07        RLCA
1D6B 4F        LD   C,A
1D6C 0600      LD   B,00H
1D6E EB        EX   DE,HL
1D6F 212218    LD   HL,1822H
1D72 09        ADD  HL,BC
1D73 4E        LD   C,(HL)
1D74 23        INC  HL
1D75 46        LD   B,(HL)
1D76 C5        PUSH BC
1D77 EB        EX   DE,HL
1D78 23        INC  HL
1D79 7E        LD   A,(HL)
1D7A FE3A      CP   3AH
1D7C D0        RET  NC
1D7D FE20      CP   20H
1D7F CA781D    JP   Z,1D78H
1D82 FE0B      CP   0BH
1D84 3005      JR   NC,1D8BH
1D86 FE09      CP   09H
1D88 D2781D    JP   NC,1D78H
1D8B FE30      CP   30H
1D8D 3F        CCF
1D8E 3C        INC  A
1D8F 3D        DEC  A
1D90 C9        RET
1D91 EB        EX   DE,HL
1D92 2AA440    LD   HL,(40A4H)
1D95 2B        DEC  HL
1D96 22FF40    LD   (40FFH),HL
1D99 EB        EX   DE,HL
1D9A
1D9B CD5803    CALL 0358H
1D9E B7        OR   A
1D9F
1DA0 FE60      CP   60H
1DA2 CC8403    CALL Z,0384H
1DA5 329940    LD   (4099H),A
1DA8 3D        DEC  A
1DA9 C0        RET  NZ
1DAA 3C        INC  A
1DAB C3B41D    JP   1DB4H
1DAE C0        RET  NZ
1DAF F5        PUSH AF
1DB0 CCBB41    CALL Z,41BBH
1DB3 F1        POP  AF
1DB4 22E640    LD   (40E6H),HL
1DB7 21B540    LD   HL,40B5H
1DBA 22B340    LD   (40B3H),HL
1DBD 21F6FF    LD   HL,0FFF6H
1DC0 C1        POP  BC
1DC1 2AA240    LD   HL,(40A2H)
1DC4 E5        PUSH HL
1DC5 F5        PUSH AF
1DC6 7D        LD   A,L
1DC7 A4        AND  H
1DC8 3C        INC  A
1DC9 2809      JR   Z,1DD4H
1DCB 22F540    LD   (40F5H),HL
1DCE 2AE640    LD   HL,(40E6H)
1DD1 22F740    LD   (40F7H),HL
1DD4 CD8B03    CALL 038BH
1DD7 CDF920    CALL 20F9H
1DDA F1        POP  AF
1DDB 213019    LD   HL,1930H
1DDE C2061A    JP   NZ,1A06H
1DE1 C3181A    JP   1A18H
1DE4 2AF740    LD   HL,(40F7H)
1DE7 7C        LD   A,H
1DE8 B5        OR   L
1DE9 1E20      LD   E,20H
1DEB CAA219    JP   Z,19A2H
1DEE EB        EX   DE,HL
1DEF 2AF540    LD   HL,(40F5H)
1DF2 22A240    LD   (40A2H),HL
1DF5 EB        EX   DE,HL
1DF6 C9        RET
1DF7 3EAF      LD   A,0AFH
1DF9 321B41    LD   (411BH),A
1DFC C9        RET
1DFD F1        POP  AF
1DFE E1        POP  HL
1DFF C9        RET
1E00 1E03      LD   E,03H
1E02 011E02    LD   BC,021EH
1E05 011E04    LD   BC,041EH
1E08 011E08    LD   BC,081EH
1E0B CD3D1E    CALL 1E3DH
```

```
1E0E 019719     LD   BC,1997H
1E11 C5         PUSH BC
1E12 D8         RET  C
1E13 D641       SUB  41H
1E15 4F         LD   C,A
1E16 47         LD   B,A
1E17 D7         RST  10H
1E18 FECE       CP   0CEH
1E1A 2009       JR   NZ,1E25H
1E1C D7         RST  10H
1E1D CD3D1E     CALL 1E3DH
1E20 D8         RET  C
1E21 D641       SUB  41H
1E23 47         LD   B,A
1E24
1E25 78         LD   A,B
1E26 91         SUB  C
1E27 D8         RET  C
1E28 3C         INC  A
1E29 E3         EX   (SP),HL
1E2A 210141     LD   HL,4101H
1E2D 0600       LD   B,00H
1E2F 09         ADD  HL,BC
1E30 73         LD   (HL),E
1E31 23         INC  HL
1E32 3D         DEC  A
1E33 20FB       JR   NZ,1E30H
1E35 E1         POP  HL
1E36 7E         LD   A,(HL)
1E37 FE2C       CP   2CH
1E39 C0         RET  NZ
1E3A D7         RST  10H
1E3B 18CE       JR   1E0BH
1E3D 7E         LD   A,(HL)
1E3E FE41       CP   41H
1E40 D8         RET  C
1E41 FE5B       CP   5BH
1E43 3F         CCF
1E44 C9         RET
1E45 D7         RST  10H
1E46 CD022B     CALL 2B02H
1E49 F0         RET  P
1E4A 1E08       LD   E,08H
1E4C C3A219     JP   19A2H
1E4F 7E         LD   A,(HL)
1E50 FE2E       CP   2EH
1E52 EB         EX   DE,HL
1E53 2AEC40     LD   HL,(40ECH)
1E56 EB         EX   DE,HL
1E57 CA781D     JP   Z,1D78H
1E5A 2B         DEC  HL
1E5B 110000     LD   DE,0000H
1E5E
1E5F D0         RET  NC
1E60 E5         PUSH HL
1E61 F5         PUSH AF
1E62 219819     LD   HL,1998H
1E65 DF         RST  18H
1E66 DA9719     JP   C,1997H
1E69 62         LD   H,D
1E6A 6B         LD   L,E
1E6B 19         ADD  HL,DE
1E6C 29         ADD  HL,HL
1E6D 19         ADD  HL,DE
1E6E 29         ADD  HL,HL
1E6F F1         POP  AF
1E70 D630       SUB  30H
1E72 5F         LD   E,A
1E73 1600       LD   D,00H
1E75 19         ADD  HL,DE
1E76 EB         EX   DE,HL
1E77 E1         POP  HL
1E78 18E4       JR   1E5EH
1E7A CA611B     JP   Z,1B61H
1E7D CD461E     CALL 1E46H
1E80 2B         DEC  HL
1E81
1E82 C0         RET  NZ
1E83 E5         PUSH HL
1E84 2AB140     LD   HL,(40B1H)
1E87 7D         LD   A,L
1E88 93         SUB  E
1E89 5F         LD   E,A
1E8A 7C         LD   A,H
1E8B 9A         SBC  A,D
1E8C 57         LD   D,A
1E8D DA7A19     JP   C,197AH
1E90 2AF940     LD   HL,(40F9H)
1E93 012800     LD   BC,0028H
1E96 09         ADD  HL,BC
1E97
1E98 D27A19     JP   NC,197AH
1E9B EB         EX   DE,HL
1E9C 22A040     LD   (40A0H),HL
1E9F E1         POP  HL
1EA0 C3611B     JP   1B61H
1EA3 CA5D1B     JP   Z,1B5DH
1EA6 CDC741     CALL 41C7H
1EA9 CD611B     CALL 1B61H
1EAC 011E1D     LD   BC,1D1EH
1EAF 1810       JR   1EC1H
1EB1 0E03       LD   C,03H
1EB3 CD6319     CALL 1963H
1EB6 C1         POP  BC
1EB7 E5         PUSH HL
1EB8 E5         PUSH HL
1EB9 2AA240     LD   HL,(40A2H)
1EBC E3         EX   (SP),HL
1EBD 3E91       LD   A,91H
1EBF F5         PUSH AF
1EC0 33         INC  SP
1EC1 C5         PUSH BC
1EC2 CD5A1E     CALL 1E5AH
1EC5 CD071F     CALL 1F07H
1EC8 E5         PUSH HL
1EC9 2AA240     LD   HL,(40A2H)
1ECC DF         RST  18H
1ECD E1         POP  HL
1ECE 23         INC  HL
1ECF DC2F1B     CALL C,1B2FH
```

```
1ED2 D42C1B     CALL NC,1B2CH
1ED5 60         LD   H,B
1ED6 69         LD   L,C
1ED7 2B         DEC  HL
1ED8 D8         RET  C
1ED9 1E0E       LD   E,0EH
1EDB C3A219     JP   19A2H
1EDE
1EDF 16FF       LD   D,0FFH
1EE1 CD3619     CALL 1936H
1EE4 F9         LD   SP,HL
1EE5 22E840     LD   (40E8H),HL
1EE8 FE91       CP   91H
1EEA 1E04       LD   E,04H
1EEC C2A219     JP   NZ,19A2H
1EEF E1         POP  HL
1EF0 22A240     LD   (40A2H),HL
1EF3 23         INC  HL
1EF4 7C         LD   A,H
1EF5 B5         OR   L
1EF6 2007       JR   NZ,1EFFH
1EF8 3ADD40     LD   A,(40DDH)
1EFB B7         OR   A
1EFC C2181A     JP   NZ,1A18H
1EFF 211E1D     LD   HL,1D1EH
1F02 E3         EX   (SP),HL
1F03 3EE1       LD   A,0E1H
1F05 013A0E     LD   BC,0E3AH
1F08 00         NOP
1F09 0600       LD   B,00H
1F0B 79         LD   A,C
1F0C 48         LD   C,B
1F0D 47         LD   B,A
1F0E 7E         LD   A,(HL)
1F0F B7         OR   A
1F10 C8         RET  Z
1F11 B8         CP   B
1F12
1F13 23         INC  HL
1F14 FE22       CP   22H
1F16 28F3       JR   Z,1F0BH
1F18 D68F       SUB  8FH
1F1A 20F2       JR   NZ,1F0EH
1F1C B8         CP   B
1F1D 8A         ADC  A,D
1F1E 57         LD   D,A
1F1F 18ED       JR   1F0EH
1F21 CD0D26     CALL 260DH
1F24 CF         RST  08H
1F25 D5         PUSH DE
1F26 EB         EX   DE,HL
1F27 22DF40     LD   (40DFH),HL
1F2A EB         EX   DE,HL
1F2B D5         PUSH DE
1F2C E7         RST  20H
1F2D F5         PUSH AF
1F2E CD3723     CALL 2337H
1F31 F1         POP  AF
1F32 E3         EX   (SP),HL
1F33 C603       ADD  03H
1F35 CD1928     CALL 2819H
1F38 CD030A     CALL 0A03H
1F3B E5         PUSH HL
1F3C 2028       JR   NZ,1F66H
1F3E 2A2141     LD   HL,(4121H)
1F41 E5         PUSH HL
1F42 23         INC  HL
1F43 5E         LD   E,(HL)
1F44 23         INC  HL
1F45 56         LD   D,(HL)
1F46 2AA440     LD   HL,(40A4H)
1F49 DF         RST  18H
1F4A 300E       JR   NC,1F5AH
1F4C 2AA040     LD   HL,(40A0H)
1F4F DF         RST  18H
1F50 D1         POP  DE
1F51 300F       JR   NC,1F62H
1F53 2AF940     LD   HL,(40F9H)
1F56 DF         RST  18H
1F57 3009       JR   NC,1F62H
1F59 3ED1       LD   A,0D1H
1F5B CDF529     CALL 29F5H
1F5E EB         EX   DE,HL
1F5F CD4328     CALL 2843H
1F62 CDF529     CALL 29F5H
1F65 E3         EX   (SP),HL
1F66 CDD309     CALL 09D3H
1F69 D1         POP  DE
1F6A E1         POP  HL
1F6B C9         RET
1F6C FE9E       CP   9EH
1F6E 2025       JR   NZ,1F95H
1F70 D7         RST  10H
1F71 CF         RST  08H
1F72 8D         ADC  A,L
1F73 CD5A1E     CALL 1E5AH
1F76 7A         LD   A,D
1F77 B3         OR   E
1F78 2809       JR   Z,1F83H
1F7A CD2A1B     CALL 1B2AH
1F7D 50         LD   D,B
1F7E 59         LD   E,C
1F7F E1         POP  HL
1F80 D2D91E     JP   NC,1ED9H
1F83 EB         EX   DE,HL
1F84 22F040     LD   (40F0H),HL
1F87 EB         EX   DE,HL
1F88
1F89 3AF240     LD   A,(40F2H)
1F8C B7         OR   A
1F8D
1F8E 3A9A40     LD   A,(409AH)
1F91 5F         LD   E,A
1F92 C3AB19     JP   19ABH
1F95 CD1C2B     CALL 2B1CH
1F98 7E         LD   A,(HL)
1F99 47         LD   B,A
1F9A FE91       CP   91H
1F9C 2803       JR   Z,1FA1H
1F9E CF         RST  08H
```

```
1F9F 8D        ADC   A,L
1FA0 2B        DEC   HL
1FA1 4B        LD    C,E
1FA2 0D        DEC   C
1FA3 78        LD    A,B
1FA4 CA601D    JP    Z,1D60H
1FA7 CD5B1E    CALL  1E5BH
1FAA FE2C      CP    2CH
1FAC C0        RET   NZ
1FAD 18F3      JR    1FA2H
1FAF 11F240    LD    DE,40F2H
1FB2 1A        LD    A,(DE)
1FB3 B7        OR    A
1FB4 CAA019    JP    Z,19A0H
1FB7
1FB8 329A40    LD    (409AH),A
1FBB 12        LD    (DE),A
1FBC 7E        LD    A,(HL)
1FBD FE87      CP    87H
1FBF 280C      JR    Z,1FCDH
1FC1 CD5A1E    CALL  1E5AH
1FC4 C0        RET   NZ
1FC5 7A        LD    A,D
1FC6 B3        OR    E
1FC7 C2C51E    JP    NZ,1EC5H
1FCA 3C        INC   A
1FCB 1802      JR    1FCFH
1FCD D7        RST   10H
1FCE C0        RET   NZ
1FCF 2AEE40    LD    HL,(40EEH)
1FD2 EB        EX    DE,HL
1FD3 2AEA40    LD    HL,(40EAH)
1FD6 22A240    LD    (40A2H),HL
1FD9 EB        EX    DE,HL
1FDA C0        RET   NZ
1FDB 7E        LD    A,(HL)
1FDC
1FDD 2004      JR    NZ,1FE3H
1FDF 23        INC   HL
1FE0 23        INC   HL
1FE1 23        INC   HL
1FE2 23        INC   HL
1FE3 23        INC   HL
1FE4 7A        LD    A,D
1FE5 A3        AND   E
1FE6 3C        INC   A
1FE7 C2051F    JP    NZ,1F05H
1FEA 3ADD40    LD    A,(40DDH)
1FED 3D        DEC   A
1FEE CABE1D    JP    Z,1DBEH
1FF1 C3051F    JP    1F05H
1FF4 CD1C2B    CALL  2B1CH
1FF7 C0        RET   NZ
1FF8 B7        OR    A
1FF9 CA4A1E    JP    Z,1E4AH
1FFC 3D        DEC   A
1FFD 87        ADD   A,A
1FFE 5F        LD    E,A
1FFF FE2D      CP    2DH
2001 3802      JR    C,2005H
2003 1E26      LD    E,26H
2005 C3A219    JP    19A2H
2008 1m0A00    LD    DE,000AH
200B D5        PUSH  DE
200C 2817      JR    Z,2025H
200E CD4F1E    CALL  1E4FH
2011
2012 E3        EX    (SP),HL
2013 2811      JR    Z,2026H
2015
2016 CF        RST   08H
2017 2C        INC   L
2018 EB        EX    DE,HL
2019 2AE440    LD    HL,(40E4H)
201C EB        EX    DE,HL
201D 2806      JR    Z,2025H
201F CD5A1E    CALL  1E5AH
2022 C29719    JP    NZ,1997H
2025 EB        EX    DE,HL
2026 7C        LD    A,H
2027 B5        OR    L
2028 CA4A1E    JP    Z,1E4AH
202B 22E440    LD    (40E4H),HL
202E 32E140    LD    (40E1H),A
2031 E1        POP   HL
2032 22E240    LD    (40E2H),HL
2035 C1        POP   BC
2036 C3331A    JP    1A33H
2039 CD3723    CALL  2337H
203C 7E        LD    A,(HL)
203D FE2C      CP    2CH
203F CC781D    CALL  Z,1D78H
2042 FECA      CP    0CAH
2044 CC781D    CALL  Z,1D78H
2047 2B        DEC   HL
2048 E5        PUSH  HL
2049 CD9409    CALL  0994H
204C E1        POP   HL
204D 2807      JR    Z,2056H
204F D7        RST   10H
2050 DAC21E    JP    C,1EC2H
2053 C35F1D    JP    1D5FH
2056 1601      LD    D,01H
2058 CD051F    CALL  1F05H
205B B7        OR    A
205C C8        RET   Z
205D D7        RST   10H
205E FE95      CP    95H
2060 20F6      JR    NZ,2058H
2062 15        DEC   D
2063 20F3      JR    NZ,2058H
2065 18E8      JR    204FH
2067 3E01      LD    A,01H
2069 329C40    LD    (409CH),A
206C C39B20    JP    209BH
206F CDCA41    CALL  41CAH
2072 FE40      CP    40H
2074 2019      JR    NZ,208FH
2076 CD012B    CALL  2B01H
2079 FE04      CP    04H
```

```
207B D24A1E     JP    NC,1E4AH
207E E5         PUSH  HL
207F 21003C     LD    HL,3C00H
2082 19         ADD   HL,DE
2083 222040     LD    (4020H),HL
2086 7B         LD    A,E
2087 E63F       AND   3FH
2089 32A640     LD    (40A6H),A
208C E1         POP   HL
208D CF         RST   08H
208E 2C         INC   L
208F FE23       CP    23H
2091 2008       JR    NZ,209BH
2093 CD8402     CALL  0284H
2096 3E80       LD    A,80H
2098 329C40     LD    (409CH),A
209B 2B         DEC   HL
209C
209D CCFE20     CALL  Z,20FEH
20A0 CA6921     JP    Z,2169H
20A3 FEBF       CP    0BFH
20A5 CABD2C     JP    Z,2CBDH
20A8 FEBC       CP    0BCH
20AA CA3721     JP    Z,2137H
20AD E5         PUSH  HL
20AE FE2C       CP    2CH
20B0 CA0821     JP    Z,2108H
20B3 FE3B       CP    3BH
20B5 CA6421     JP    Z,2164H
20B8 C1         POP   BC
20B9 CD3723     CALL  2337H
20BC E5         PUSH  HL
20BD
20BE 2832       JR    Z,20F2H
20C0 CDBD0F     CALL  0FBDH
20C3 CD6528     CALL  2865H
20C6 CDCD41     CALL  41CDH
20C9 2A2141     LD    HL,(4121H)
20CC 3A9C40     LD    A,(409CH)
20CF B7         OR    A
20D0 FAE920     JP    M,20E9H
20D3 2808       JR    Z,20DDH
20D5 3A9B40     LD    A,(409BH)
20D8 86         ADD   A,(HL)
20D9 FE84       CP    84H
20DB 1809       JR    20E6H
20DD 3A9D40     LD    A,(409DH)
20E0 47         LD    B,A
20E1 3AA640     LD    A,(40A6H)
20E4 86         ADD   A,(HL)
20E5 B8         CP    B
20E6 D4FE20     CALL  NC,20FEH
20E9 CDAA28     CALL  28AAH
20EC 3E20       LD    A,20H
20EE CD2A03     CALL  032AH
20F1 B7         OR    A
20F2 CCAA28     CALL  Z,28AAH
20F5 E1         POP   HL
20F6 C39B20     JP    209BH
20F9 3AA640     LD    A,(40A6H)
20FC B7         OR    A
20FD
20FE 3E0D       LD    A,0DH
2100 CD2A03     CALL  032AH
2103 CDD041     CALL  41D0H
2106 AF         XOR   A
2107
2108 CDD341     CALL  41D3H
210B 3A9C40     LD    A,(409CH)
210E B7         OR    A
210F F21921     JP    P,2119H
2112 3E2C       LD    A,2CH
2114 CD2A03     CALL  032AH
2117 184B       JR    2164H
2119 2808       JR    Z,2123H
211B 3A9B40     LD    A,(409BH)
211E FE70       CP    70H
2120 C32B21     JP    212BH
2123 3A9E40     LD    A,(409EH)
2126 47         LD    B,A
2127 3AA640     LD    A,(40A6H)
212A B8         CP    B
212B D4FE20     CALL  NC,20FEH
212E 3034       JR    NC,2164H
2130 D610       SUB   10H
2132 30FC       JR    NC,2130H
2134 2F         CPL
2135 1823       JR    215AH
2137 CD1B2B     CALL  2B1BH
213A E63F       AND   3FH
213C 5F         LD    E,A
213D CF         RST   08H
213E 29         ADD   HL,HL
213F 2B         DEC   HL
2140 E5         PUSH  HL
2141 CDD341     CALL  41D3H
2144 3A9C40     LD    A,(409CH)
2147 B7         OR    A
2148 FA4A1E     JP    M,1E4AH
214B CA5321     JP    Z,2153H
214E 3A9B40     LD    A,(409BH)
2151 1803       JR    2156H
2153 3AA640     LD    A,(40A6H)
2156 2F         CPL
2157 83         ADD   A,E
2158 300A       JR    NC,2164H
215A 3C         INC   A
215B 47         LD    B,A
215C 3E20       LD    A,20H
215E CD2A03     CALL  032AH
2161 05         DEC   B
2162 20FA       JR    NZ,215EH
2164 E1         POP   HL
2165 D7         RST   10H
2166 C3A020     JP    20A0H
2169 3A9C40     LD    A,(409CH)
216C B7         OR    A
216D FCF801     CALL  M,01F8H
2170 AF         XOR   A
2171 329C40     LD    (409CH),A
```

```
2174 CDBE41    CALL 41BEH
2177
2178 3F        CCF
2179 52        LD   D,D
217A 45        LD   B,L
217B 44        LD   B,H
217C 4F        LD   C,A
217D 0D        DEC  C
217E 00        NOP
217F 3ADE40    LD   A,(40DEH)
2182 B7        OR   A
2183 C29119    JP   NZ,1991H
2186 3AA940    LD   A,(40A9H)
2189 B7        OR   A
218A 1E2A      LD   E,2AH
218C CAA219    JP   Z,19A2H
218F C1        POP  BC
2190 217821    LD   HL,2178H
2193 CDA728    CALL 28A7H
2196 2AE640    LD   HL,(40E6H)
2199 C9        RET
219A CD2828    CALL 2828H
219D 7E        LD   A,(HL)
219E CDD641    CALL 41D6H
21A1 D623      SUB  23H
21A3 32A940    LD   (40A9H),A
21A6 7E        LD   A,(HL)
21A7 2020      JR   NZ,21C9H
21A9 CD9302    CALL 0293H
21AC E5        PUSH HL
21AD 06FA      LD   B,0FAH
21AF 2AA740    LD   HL,(40A7H)
21B2 CD3502    CALL 0235H
21B5 77        LD   (HL),A
21B6 23        INC  HL
21B7 FE0D      CP   0DH
21B9 2802      JR   Z,21BDH
21BB 10F5      DJNZ 21B2H
21BD 2B        DEC  HL
21BE 3600      LD   (HL),00H
21C0 CDF801    CALL 01F8H
21C3 2AA740    LD   HL,(40A7H)
21C6 2B        DEC  HL
21C7 1822      JR   21EBH
21C9 01DB21    LD   BC,21DBH
21CC C5        PUSH BC
21CD FE22      CP   22H
21CF C0        RET  NZ
21D0 CD6628    CALL 2866H
21D3 CF        RST  08H
21D4 3B        DEC  SP
21D5 E5        PUSH HL
21D6 CDAA28    CALL 28AAH
21D9 E1        POP  HL
21DA
21DB E5        PUSH HL
21DC CDB31B    CALL 1BB3H
21DF C1        POP  BC
21E0 DABE1D    JP   C,1DBEH
21E3 23        INC  HL
21E4 7E        LD   A,(HL)
21E5 B7        OR   A
21E6 2B        DEC  HL
21E7 C5        PUSH BC
21E8 CA041F    JP   Z,1F04H
21EB 362C      LD   (HL),2CH
21ED 1805      JR   21F4H
21EF E5        PUSH HL
21F0 2AFF40    LD   HL,(40FFH)
21F3 F6AF      OR   0AFH
21F5 32DE40    LD   (40DEH),A
21F8 E3        EX   (SP),HL
21F9 1802      JR   21FDH
21FB CF        RST  08H
21FC 2C        INC  L
21FD CD0D26    CALL 260DH
2200
2201 D5        PUSH DE
2202 7E        LD   A,(HL)
2203 FE2C      CP   2CH
2205 2826      JR   Z,222DH
2207 3ADE40    LD   A,(40DEH)
220A B7        OR   A
220B C29622    JP   NZ,2296H
220E 3AA940    LD   A,(40A9H)
2211 B7        OR   A
2212 1E06      LD   E,06H
2214 CAA219    JP   Z,19A2H
2217 3E3F      LD   A,3FH
2219 CD2A03    CALL 032AH
221C CDB31B    CALL 1BB3H
221F D1        POP  DE
2220 C1        POP  BC
2221 DABE1D    JP   C,1DBEH
2224 23        INC  HL
2225 7E        LD   A,(HL)
2226 B7        OR   A
2227 2B        DEC  HL
2228 C5        PUSH BC
2229 CA041F    JP   Z,1F04H
222C D5        PUSH DE
222D CDDC41    CALL 41DCH
2230 E7        RST  20H
2231 F5        PUSH AF
2232 2019      JR   NZ,224DH
2234 D7        RST  10H
2235 57        LD   D,A
2236 47        LD   B,A
2237 FE22      CP   22H
2239 2805      JR   Z,2240H
223B 163A      LD   D,3AH
223D 062C      LD   B,2CH
223F 2B        DEC  HL
2240 CD6928    CALL 2869H
2243 F1        POP  AF
2244 EB        EX   DE,HL
2245 215A22    LD   HL,225AH
2248
2249 D5        PUSH DE
224A C3331F    JP   1F33H
```

```
224D D7        RST  10H
224E F1        POP  AF
224F F5        PUSH AF
2250 014322    LD   BC,2243H
2253 C5        PUSH BC
2254 DA6C0E    JP   C,0E6CH
2257 D2650E    JP   NC,0E65H
225A 2B        DEC  HL
225B D7        RST  10H
225C 2805      JR   Z,2263H
225E FE2C      CP   2CH
2260 C27F21    JP   NZ,217FH
2263 E3        EX   (SP),HL
2264 2B
2265 D7        RST  10H
2266 C2FB21    JP   NZ,21FBH
2269 D1        POP  DE
226A 00        NOP
226B 00        NOP
226C 00        NOP
226D 00        NOP
226E 00        NOP
226F 3ADE40    LD   A,(40DEH)
2272 B7        OR   A
2273 EB        EX   DE,HL
2274 C2961D    JP   NZ,1D96H
2277 D5        PUSH DE
2278 CDDF41    CALL 41DFH
227B B6        OR   (HL)
227C 218622    LD   HL,2286H
227F C4A728    CALL NZ,28A7H
2282 E1        POP  HL
2283 C36921    JP   2169H
2286 3F        CCF
2287 45        LD   B,L
2288 78        LD   A,B
2289 74        LD   (HL),H
228A 72        LD   (HL),D
228B 61        LD   H,C
228C 2069      JR   NZ,22F7H
228E 67        LD   H,A
228F 6E        LD   L,(HL)
2290 6F        LD   L,A
2291 72        LD   (HL),D
2292 65        LD   H,L
2293 64        LD   H,H
2294 0D        DEC  C
2295 00        NOP
2296 CD051F    CALL 1F05H
2299 B7        OR   A
229A 2012      JR   NZ,22AEH
229C 23
229D 7E        LD   A,(HL)
229E 23        INC  HL
229F B6        OR   (HL)
22A0 1E06      LD   E,06H
22A2 CAA219    JP   Z,19A2H
22A5 23        INC  HL
22A6 5E        LD   E,(HL)
22A7 23        INC  HL
22A8 56        LD   D,(HL)
22A9 EB        EX   DE,HL
22AA 22DA40    LD   (40DAH),HL
22AD EB        EX   DE,HL
22AE
22AF FE88      CP   88H
22B1 20E3      JR   NZ,2296H
22B3 C32D22    JP   222DH
22B6 110000    LD   DE,0000H
22B9 C40D26    CALL NZ,260DH
22BC 22DF40    LD   (40DFH),HL
22BF CD3619    CALL 1936H
22C2 C29D19    JP   NZ,199DH
22C5 F9        LD   SP,HL
22C6 22E840    LD   (40E8H),HL
22C9 D5        PUSH DE
22CA 7E        LD   A,(HL)
22CB 23        INC  HL
22CC F5        PUSH AF
22CD D5        PUSH DE
22CE 7E        LD   A,(HL)
22CF 23        INC  HL
22D0 B7        OR   A
22D1 FAEA22    JP   M,22EAH
22D4 CDB109    CALL 09B1H
22D7 E3        EX   (SP),HL
22D8 E5        PUSH HL
22D9 CD0B07    CALL 070BH
22DC E1        POP  HL
22DD CDCB09    CALL 09CBH
22E0 E1        POP  HL
22E1 CDC209    CALL 09C2H
22E4 E5        PUSH HL
22E5 CD0C0A    CALL 0A0CH
22E8 1829      JR   2313H
22EA 23        INC  HL
22EB 23        INC  HL
22EC 23        INC  HL
22ED 23        INC  HL
22EE 4E        LD   C,(HL)
22EF 23        INC  HL
22F0 46        LD   B,(HL)
22F1 23        INC  HL
22F2 E3        EX   (SP),HL
22F3 5E        LD   E,(HL)
22F4 23        INC  HL
22F5 56        LD   D,(HL)
22F6 E5        PUSH HL
22F7 69        LD   L,C
22F8 60        LD   H,B
22F9 CDD20B    CALL 0BD2H
22FC 3AAF40    LD   A,(40AFH)
22FF FE04      CP   04H
2301 CAB207    JP   Z,07B2H
2304
2305 E1        POP  HL
2306 72        LD   (HL),D
2307 2B        DEC  HL
2308 73        LD   (HL),E
2309 E1        POP  HL
```

```
230A D5        PUSH DE
230B 5E        LD   E,(HL)
230C 23        INC  HL
230D 56        LD   D,(HL)
230E 23        INC  HL
230F E3        EX   (SP),HL
2310 CD390A    CALL 0A39H
2313 E1        POP  HL
2314 C1        POP  BC
2315 90        SUB  B
2316 CDC209    CALL 09C2H
2319 2809      JR   Z,2324H
231B EB        EX   DE,HL
231C 22A240    LD   (40A2H),HL
231F 69        LD   L,C
2320 60        LD   H,B
2321 C31A1D    JP   1D1AH
2324 F9        LD   SP,HL
2325 22E840    LD   (40E8H),HL
2328 2ADF40    LD   HL,(40DFH)
232B 7E        LD   A,(HL)
232C FE2C      CP   2CH
232E C21E1D    JP   NZ,1D1EH
2331 D7        RST  10H
2332 CDB922    CALL 22B9H
2335 CF        RST  08H
2336 282B      JR   Z,2363H
2338 1600      LD   D,00H
233A D5        PUSH DE
233B 0E01      LD   C,01H
233D CD6319    CALL 1963H
2340 CD9F24    CALL 249FH
2343 22F340    LD   (40F3H),HL
2346 2AF340    LD   HL,(40F3H)
2349 C1        POP  BC
234A 7E        LD   A,(HL)
234B 1600      LD   D,00H
234D D6D4      SUB  0D4H
234F 3813      JR   C,2364H
2351 FE03      CP   03H
2353 300F      JR   NC,2364H
2355 FE01      CP   01H
2357 17        RLA
2358 AA        XOR  D
2359 BA        CP   D
235A 57        LD   D,A
235B DA9719    JP   C,1997H
235E 22D840    LD   (40D8H),HL
2361 D7        RST  10H
2362 18E9      JR   234DH
2364 7A        LD   A,D
2365 B7        OR   A
2366 C2EC23    JP   NZ,23ECH
2369 7E        LD   A,(HL)
236A 22D840    LD   (40D8H),HL
236D D6CD      SUB  0CDH
236F
2370 FE07      CP   07H
2372
2373 5F        LD   E,A
2374 3AAF40    LD   A,(40AFH)
2377 D603      SUB  03H
2379 B3        OR   E
237A CA8F29    JP   Z,298FH
237D 219A18    LD   HL,189AH
2380 19        ADD  HL,DE
2381 78        LD   A,B
2382 56        LD   D,(HL)
2383 BA        CP   D
2384
2385 C5        PUSH BC
2386 014623    LD   BC,2346H
2389 C5        PUSH BC
238A 7A        LD   A,D
238B FE7F      CP   7FH
238D CAD423    JP   Z,23D4H
2390 FE51      CP   51H
2392 DAE123    JP   C,23E1H
2395 212141    LD   HL,4121H
2398 B7        OR   A
2399 3AAF40    LD   A,(40AFH)
239C 3D        DEC  A
239D 3D        DEC  A
239E 3D        DEC  A
239F CAF60A    JP   Z,0AF6H
23A2 4E        LD   C,(HL)
23A3 23        INC  HL
23A4 46        LD   B,(HL)
23A5 C5        PUSH BC
23A6 FAC523    JP   M,23C5H
23A9 23        INC  HL
23AA 4E        LD   C,(HL)
23AB 23        INC  HL
23AC 46        LD   B,(HL)
23AD C5        PUSH BC
23AE F5        PUSH AF
23AF B7        OR   A
23B0 E2C423    JP   PO,23C4H
23B3 F1        POP  AF
23B4 23        INC  HL
23B5 3803      JR   C,23BAH
23B7 211D41    LD   HL,411DH
23BA 4E        LD   C,(HL)
23BB 23        INC  HL
23BC 46        LD   B,(HL)
23BD
23BE C5        PUSH BC
23BF 4E        LD   C,(HL)
23C0 23        INC  HL
23C1 46        LD   B,(HL)
23C2 C5        PUSH BC
23C3 06F1      LD   B,0F1H
23C5 C603      ADD  03H
23C7 4B        LD   C,E
23C8 47        LD   B,A
23C9 C5        PUSH BC
23CA 010624    LD   BC,2406H
23CD C5        PUSH BC
23CE 2AD840    LD   HL,(40D8H)
23D1 C33A23    JP   233AH
```

```
23D4 CDB10A     CALL 0AB1H
23D7 CDA409     CALL 09A4H
23DA 01F213     LD   BC,13F2H
23DD 167F       LD   D,7FH
23DF 18EC       JR   23CDH
23E1 D5         PUSH DE
23E2 CD7F0A     CALL 0A7FH
23E5 D1         POP  DE
23E6 E5         PUSH HL
23E7 01E925     LD   BC,25E9H
23EA 18E1       JR   23CDH
23EC 78         LD   A,B
23ED FE64       CP   64H
23EF
23F0 C5         PUSH BC
23F1 D5         PUSH DE
23F2 110464     LD   DE,6404H
23F5 21B825     LD   HL,25B8H
23F8 E5         PUSH HL
23F9 E7         RST  20H
23FA C29523     JP   NZ,2395H
23FD 2A2141     LD   HL,(4121H)
2400 E5         PUSH HL
2401 018C25     LD   BC,258CH
2404 18C7       JR   23CDH
2406 C1         POP  BC
2407 79         LD   A,C
2408 32B040     LD   (40B0H),A
240B 78         LD   A,B
240C FE08       CP   08H
240E 2828       JR   Z,2438H
2410 3AAF40     LD   A,(40AFH)
2413 FE08       CP   08H
2415 CA6024     JP   Z,2460H
2418 57         LD   D,A
2419 78         LD   A,B
241A FE04       CP   04H
241C CA7224     JP   Z,2472H
241F 7A         LD   A,D
2420 FE03       CP   03H
2422 CAF60A     JP   Z,0AF6H
2425 D27C24     JP   NC,247CH
2428 21BF18     LD   HL,18BFH
242B 0600       LD   B,00H
242D 09         ADD  HL,BC
242E 09         ADD  HL,BC
242F 4E         LD   C,(HL)
2430 23         INC  HL
2431 46         LD   B,(HL)
2432 D1         POP  DE
2433 2A2141     LD   HL,(4121H)
2436 C5         PUSH BC
2437
2438 CDDB0A     CALL 0ADBH
243B CDF<09     CALL 09FCH
243E E1         POP  HL
243F 221F41     LD   (411FH),HL
2442 E1         POP  HL
2443 221D41     LD   (411DH),HL
2446 C1         POP  BC
2447 D1
2448 CDB409     CALL 09B4H
244B CDDB0A     CALL 0ADBH
244E 21AB18     LD   HL,18ABH
2451 3AB040     LD   A,(40B0H)
2454 07         RLCA
2455 C5         PUSH BC
2456 4F         LD   C,A
2457 0600       LD   B,00H
2459 09         ADD  HL,BC
245A C1         POP  BC
245B 7E         LD   A,(HL)
245C 23         INC  HL
245D 66         LD   H,(HL)
245E 6F         LD   L,A
245F E9         JP   (HL)
2460 C5         PUSH BC
2461 CDFC09     CALL 09FCH
2464 F1         POP  AF
2465 32AF40     LD   (40AFH),A
2468 FE04       CP   04H
246A 28DA       JR   Z,2446H
246C E1         POP  HL
246D 222141     LD   (4121H),HL
2470 18D9       JR   244BH
2472 CDB10A     CALL 0AB1H
2475 C1         POP  BC
2476 D1
2477 21B518     LD   HL,18B5H
247A 18D5       JR   2451H
247C E1         POP  HL
247D CDA409     CALL 09A4H
2480 CDCF0A     CALL 0ACFH
2483 CDBF09     CALL 09BFH
2486 E1         POP  HL
2487 222341     LD   (4123H),HL
248A E1         POP  HL
248B 222141     LD   (4121H),HL
248E 18E7       JR   2477H
2490 E5         PUSH HL
2491 EB         EX   DE,HL
2492 CDCF0A     CALL 0ACFH
2495 E1         POP  HL
2496 CDA409     CALL 09A4H
2499 CDCF0A     CALL 0ACFH
249C C3A008     JP   08A0H
249F D7         RST  10H
24A0 1E28       LD   E,28H
24A2 CAA219     JP   Z,19A2H
24A5 DA6C0E     JP   C,0E6CH
24A8 CD3D1E     CALL 1E3DH
24AB D24025     JP   NC,2540H
24AE FECD       CP   0CDH
24B0 28ED       JR   Z,249FH
24B2 FE2E       CP   2EH
24B4 CA6C0E     JP   Z,0E6CH
24B7 FECE       CP   0CEH
24B9 CA3225     JP   Z,2532H
24BC FE22       CP   22H
24BE CA6628     JP   Z,2866H
```

```
24C1 FECB      CP   0CBH
24C3 CAC425    JP   Z,25C4H
24C6 FE26      CP   26H
24C8 CA9441    JP   Z,4194H
24CB FEC3      CP   0C3H
24CD 200A      JR   NZ,24D9H
24CF D7        RST  10H
24D0 3A9A40    LD   A,(409AH)
24D3 E5        PUSH HL
24D4 CDF827    CALL 27F8H
24D7 E1        POP  HL
24D8
24D9 FEC2      CP   0C2H
24DB 200A      JR   NZ,24E7H
24DD D7        RST  10H
24DE E5        PUSH HL
24DF 2AEA40    LD   HL,(40EAH)
24E2 CD660C    CALL 0C66H
24E5 E1        POP  HL
24E6 C9        RET
24E7 FEC0      CP   0C0H
24E9 2014      JR   NZ,24FFH
24EB D7        RST  10H
24EC CF        RST  08H
24ED 28CD      JR   Z,24BCH
24EF 0D        DEC  C
24F0 26CF      LD   H,0CFH
24F2 29        ADD  HL,HL
24F3 E5        PUSH HL
24F4 EB        EX   DE,HL
24F5 7C        LD   A,H
24F6 B5        OR   L
24F7 CA4A1E    JP   Z,1E4AH
24FA CD9A0A    CALL 0A9AH
24FD E1        POP  HL
24FE
24FF FEC1      CP   0C1H
2501 CAFE27    JP   Z,27FEH
2504 FEC5      CP   0C5H
2506 CA9D41    JP   Z,419DH
2509 FEC8      CP   0C8H
250B CAC927    JP   Z,27C9H
250E FEC7      CP   0C7H
2510 CA7641    JP   Z,4176H
2513 FEC6      CP   0C6H
2515 CA3201    JP   Z,0132H
2518 FEC9      CP   0C9H
251A CA9D01    JP   Z,019DH
251D FEC4      CP   0C4H
251F CA2F2A    JP   Z,2A2FH
2522 FEBE      CP   0BEH
2524 CA5541    JP   Z,4155H
2527 D6D7      SUB  0D7H
2529 D24E25    JP   NC,254EH
252C CD3523    CALL 2335H
252F CF        RST  08H
2530 29        ADD  HL,HL
2531 C9        RET
2532 167D      LD   D,7DH
2534 CD3A23    CALL 233AH
2537 2AF340    LD   HL,(40F3H)
253A E5        PUSH HL
253B CD7B09    CALL 097BH
253E E1        POP  HL
253F
2540 CD0D26    CALL 260DH
2543 E5        PUSH HL
2544 EB        EX   DE,HL
2545 222141    LD   (4121H),HL
2548 E7        RST  20H
2549 C4F709    CALL NZ,09F7H
254C E1        POP  HL
254D
254E 0600      LD   B,00H
2550 07        RLCA
2551 4F        LD   C,A
2552 C5        PUSH BC
2553 D7        RST  10H
2554 79        LD   A,C
2555 FE41      CP   41H
2557 3816      JR   C,256FH
2559 CD3523    CALL 2335H
255C CF        RST  08H
255D 2C        INC  L
255E CDF40A    CALL 0AF4H
2561 EB        EX   DE,HL
2562 2A2141    LD   HL,(4121H)
2565 E3        EX   (SP),HL
2566 E5        PUSH HL
2567 EB        EX   DE,HL
2568 CD1C2B    CALL 2B1CH
256B EB        EX   DE,HL
256C E3        EX   (SP),HL
256D 1814      JR   2583H
256F CD2C25    CALL 252CH
2572 E3        EX   (SP),HL
2573 7D        LD   A,L
2574 FE0C      CP   0CH
2576 3807      JR   C,257FH
2578 FE1B      CP   1BH
257A E5        PUSH HL
257B DCB10A    CALL C,0AB1H
257E E1        POP  HL
257F 113E25    LD   DE,253EH
2582 D5        PUSH DE
2583 010816    LD   BC,1608H
2586 09        ADD  HL,BC
2587 4E        LD   C,(HL)
2588 23        INC  HL
2589 66        LD   H,(HL)
258A 69        LD   L,C
258B E9        JP   (HL)
258C CDD729    CALL 29D7H
258F 7E        LD   A,(HL)
2590 23        INC  HL
2591 4E        LD   C,(HL)
2592 23        INC  HL
2593 46        LD   B,(HL)
2594 D1        POP  DE
2595 C5        PUSH BC
```

```
2596 F5        PUSH AF
2597 CDDE29    CALL 29DEH
259A D1        POP  DE
259B 5E        LD   E,(HL)
259C 23        INC  HL
259D 4E        LD   C,(HL)
259E 23        INC  HL
259F 46        LD   B,(HL)
25A0 E1        POP  HL
25A1 7B        LD   A,E
25A2 B2        OR   D
25A3
25A4 7A        LD   A,D
25A5 D601      SUB  01H
25A7 D8        RET  C
25A8 AF        XOR  A
25A9 BB        CP   E
25AA 3C        INC  A
25AB D0        RET  NC
25AC 15        DEC  D
25AD 1D        DEC  E
25AE 0A        LD   A,(BC)
25AF BE        CP   (HL)
25B0 23        INC  HL
25B1 03        INC  BC
25B2 28ED      JR   Z,25A1H
25B4 3F        CCF
25B5 C36009    JP   0960H
25B8 3C        INC  A
25B9 8F        ADC  A,A
25BA C1        POP  BC
25BB A0        AND  B
25BC C6FF      ADD  0FFH
25BE 9F        SBC  A,A
25BF CD8D09    CALL 098DH
25C2 1812      JR   25D6H
25C4 165A      LD   D,5AH
25C6 CD3A23    CALL 233AH
25C9 CD7F0A    CALL 0A7FH
25CC 7D        LD   A,L
25CD 2F        CPL
25CE 6F        LD   L,A
25CF 7C        LD   A,H
25D0 2F        CPL
25D1 67        LD   H,A
25D2 222141    LD   (4121H),HL
25D5 C1        POP  BC
25D6 C34623    JP   2346H
25D9 3AAF40    LD   A,(40AFH)
25DC FE08      CP   08H
25DE 3005      JR   NC,25E5H
25E0 D603      SUB  03H
25E2 B7        OR   A
25E3 37        SCF
25E4 C9        RET
25E5 D603      SUB  03H
25E7 B7        OR   A
25E8
25E9 C5        PUSH BC
25EA CD7F0A    CALL 0A7FH
25ED F1        POP  AF
25EE D1        POP  DE
25EF 01FA27    LD   BC,27FAH
25F2 C5        PUSH BC
25F3 FE46      CP   46H
25F5 2006      JR   NZ,25FDH
25F7 7B        LD   A,E
25F8 B5        OR   L
25F9 6F        LD   L,A
25FA 7C        LD   A,H
25FB B2        OR   D
25FC C9        RET
25FD 7B        LD   A,E
25FE A5        AND  L
25FF 6F        LD   L,A
2600 7C        LD   A,H
2601 A2        AND  D
2602
2603 2B        DEC  HL
2604 D7        RST  10H
2605 C8        RET  Z
2606 CF        RST  08H
2607 2C        INC  L
2608 010326    LD   BC,2603H
260B C5        PUSH BC
260C F6AF      OR   0AFH
260E 32AE40    LD   (40AEH),A
2611 46        LD   B,(HL)
2612 CD3D1E    CALL 1E3DH
2615 DA9719    JP   C,1997H
2618 AF        XOR  A
2619 4F        LD   C,A
261A D7        RST  10H
261B 3805      JR   C,2622H
261D CD3D1E    CALL 1E3DH
2620 3809      JR   C,262BH
2622 4F        LD   C,A
2623 D7        RST  10H
2624 38FD      JR   C,2623H
2626 CD3D1E    CALL 1E3DH
2629 30F8      JR   NC,2623H
262B 115226    LD   DE,2652H
262E D5        PUSH DE
262F 1602      LD   D,02H
2631 FE25      CP   25H
2633
2634 14        INC  D
2635 FE24      CP   24H
2637 C8        RET  Z
2638 14        INC  D
2639 FE21      CP   21H
263B C8        RET  Z
263C 1608      LD   D,08H
263E FE23      CP   23H
2640 C8        RET  Z
2641 78        LD   A,B
2642 D641      SUB  41H
2644 E67F      AND  7FH
2646 5F        LD   E,A
2647 1600      LD   D,00H
```

```
2649 E5        PUSH HL
264A 210141    LD   HL,4101H
264D 19        ADD  HL,DE
264E 56        LD   D,(HL)
264F E1        POP  HL
2650 2B        DEC  HL
2651
2652 7A        LD   A,D
2653 32AF40    LD   (40AFH),A
2656 D7        RST  10H
2657 3ADC40    LD   A,(40DCH)
265A B7        OR   A
265B C26426    JP   NZ,2664H
265E 7E        LD   A,(HL)
265F D628      SUB  28H
2661 CAE926    JP   Z,26E9H
2664 AF        XOR  A
2665 32DC40    LD   (40DCH),A
2668 E5        PUSH HL
2669 D5        PUSH DE
266A 2AF940    LD   HL,(40F9H)
266D EB        EX   DE,HL
266E 2AFB40    LD   HL,(40FBH)
2671 DF        RST  18H
2672 E1        POP  HL
2673 2819      JR   Z,268EH
2675 1A        LD   A,(DE)
2676 6F        LD   L,A
2677 BC        CP   H
2678 13        INC  DE
2679 200B      JR   NZ,2686H
267B 1A        LD   A,(DE)
267C B9        CP   C
267D 2007      JR   NZ,2686H
267F 13        INC  DE
2680 1A        LD   A,(DE)
2681 B8        CP   B
2682 CACC26    JP   Z,26CCH
2685 3E13      LD   A,13H
2687
2688 E5        PUSH HL
2689 2600      LD   H,00H
268B 19        ADD  HL,DE
268C 18DF      JR   266DH
268E 7C        LD   A,H
268F E1        POP  HL
2690 E3        EX   (SP),HL
2691 F5        PUSH AF
2692 D5        PUSH DE
2693 11F124    LD   DE,24F1H
2696 DF        RST  18H
2697 2836      JR   Z,26CFH
2699 114325    LD   DE,2543H
269C DF        RST  18H
269D D1        POP  DE
269E 2835      JR   Z,26D5H
26A0 F1        POP  AF
26A1 E3        EX   (SP),HL
26A2 E5        PUSH HL
26A3 C5        PUSH BC
26A4 4F        LD   C,A
26A5 0600      LD   B,00H
26A7 C5        PUSH BC
26A8 03        INC  BC
26A9 03        INC  BC
26AA 03        INC  BC
26AB 2AFD40    LD   HL,(40FDH)
26AE E5        PUSH HL
26AF 09        ADD  HL,BC
26B0 C1        POP  BC
26B1 E5        PUSH HL
26B2 CD5519    CALL 1955H
26B5 E1        POP  HL
26B6 22FD40    LD   (40FDH),HL
26B9 60        LD   H,B
26BA 69        LD   L,C
26BB 22FB40    LD   (40FBH),HL
26BE 2B        DEC  HL
26BF 3600      LD   (HL),00H
26C1 DF        RST  18H
26C2 20FA      JR   NZ,26BEH
26C4 D1        POP  DE
26C5 73        LD   (HL),E
26C6 23        INC  HL
26C7 D1        POP  DE
26C8 73        LD   (HL),E
26C9 23        INC  HL
26CA 72        LD   (HL),D
26CB EB        EX   DE,HL
26CC 13        INC  DE
26CD
26CE C9        RET
26CF 57        LD   D,A
26D0 5F        LD   E,A
26D1 F1        POP  AF
26D2 F1        POP  AF
26D3
26D4 C9        RET
26D5 322441    LD   (4124H),A
26D8 C1        POP  BC
26D9 67        LD   H,A
26DA 6F        LD   L,A
26DB 222141    LD   (4121H),HL
26DE E7        RST  20H
26DF 2006      JR   NZ,26E7H
26E1 212819    LD   HL,1928H
26E4 222141    LD   (4121H),HL
26E7 E1        POP  HL
26E8 C9        RET
26E9 E5        PUSH HL
26EA 2AAE40    LD   HL,(40AEH)
26ED E3        EX   (SP),HL
26EE 57        LD   D,A
26EF D5        PUSH DE
26F0 C5        PUSH BC
26F1 CD451E    CALL 1E45H
26F4 C1        POP  BC
26F5 F1        POP  AF
26F6 EB        EX   DE,HL
26F7 E3        EX   (SP),HL
```

```
26F8 E5        PUSH HL
26F9
26FA 3C        INC  A
26FB 57        LD   D,A
26FC 7E        LD   A,(HL)
26FD FE2C      CP   2CH
26FF 28EE      JR   Z,26EFH
2701 CF        RST  08H
2702 29        ADD  HL,HL
2703 22F340    LD   (40F3H),HL
2706 E1        POP  HL
2707 22AE40    LD   (40AEH),HL
270A D5        PUSH DE
270B 2AFB40    LD   HL,(40FBH)
270E 3E19      LD   A,19H
2710
2711 2AFD40    LD   HL,(40FDH)
2714 EB        EX   DE,HL
2715 DF        RST  18H
2716 3AAF40    LD   A,(40AFH)
2719 2827      JR   Z,2742H
271B BE        CP   (HL)
271C 23        INC  HL
271D 2008      JR   NZ,2727H
271F 7E        LD   A,(HL)
2720 B9        CP   C
2721 23        INC  HL
2722 2004      JR   NZ,2728H
2724 7E        LD   A,(HL)
2725 B8        CP   B
2726 3E23      LD   A,23H
2728 23        INC  HL
2729 5E        LD   E,(HL)
272A 23        INC  HL
272B 56        LD   D,(HL)
272C 23        INC  HL
272D 20E0      JR   NZ,270FH
272F 3AAE40    LD   A,(40AEH)
2732 B7        OR   A
2733 1E12      LD   E,12H
2735 C2A219    JP   NZ,19A2H
2738 F1        POP  AF
2739 96        SUB  (HL)
273A CA9527    JP   Z,2795H
273D 1E10      LD   E,10H
273F C3A219    JP   19A2H
2742 77        LD   (HL),A
2743 23        INC  HL
2744 5F        LD   E,A
2745 1600      LD   D,00H
2747 F1        POP  AF
2748 71        LD   (HL),C
2749 23        INC  HL
274A 70        LD   (HL),B
274B 23        INC  HL
274C 4F        LD   C,A
274D CD6319    CALL 1963H
2750 23        INC  HL
2751 23        INC  HL
2752 22D840    LD   (40D8H),HL
2755 71        LD   (HL),C
2756 23        INC  HL
2757 3AAE40    LD   A,(40AEH)
275A
275B 79        LD   A,C
275C 010B00    LD   BC,000BH
275F 3002      JR   NC,2763H
2761 C1        POP  BC
2762 03        INC  BC
2763 71        LD   (HL),C
2764 23        INC  HL
2765 70        LD   (HL),B
2766 23        INC  HL
2767 F5        PUSH AF
2768 CDAA0B    CALL 0BAAH
276B F1        POP  AF
276C 3D        DEC  A
276D 20ED      JR   NZ,275CH
276F F5        PUSH AF
2770 42        LD   B,D
2771 4B        LD   C,E
2772 EB        EX   DE,HL
2773 19        ADD  HL,DE
2774 38C7      JR   C,273DH
2776 CD6C19    CALL 196CH
2779 22FD40    LD   (40FDH),HL
277C 2B        DEC  HL
277D 3600      LD   (HL),00H
277F DF        RST  18H
2780 20FA      JR   NZ,277CH
2782 03        INC  BC
2783 57        LD   D,A
2784 2AD840    LD   HL,(40D8H)
2787 5E        LD   E,(HL)
2788 EB        EX   DE,HL
2789 29        ADD  HL,HL
278A 09        ADD  HL,BC
278B EB        EX   DE,HL
278C 2B        DEC  HL
278D 2B        DEC  HL
278E 73        LD   (HL),E
278F 23        INC  HL
2790 72        LD   (HL),D
2791 23        INC  HL
2792 F1        POP  AF
2793 3830      JR   C,27C5H
2795 47        LD   B,A
2796 4F        LD   C,A
2797 7E        LD   A,(HL)
2798 23        INC  HL
2799 16E1      LD   D,0E1H
279B 5E        LD   E,(HL)
279C 23        INC  HL
279D 56        LD   D,(HL)
279E 23        INC  HL
279F E3        EX   (SP),HL
27A0 F5        PUSH AF
27A1
27A2 D23D27    JP   NC,273DH
27A5 CDAA0B    CALL 0BAAH
```

```
27A8 19       ADD  HL,DE
27A9 F1       POP  AF
27AA 3D       DEC  A
27AB 44       LD   B,H
27AC 4D       LD   C,L
27AD 20EB     JR   NZ,279AH
27AF 3AAF40   LD   A,(40AFH)
27B2 44       LD   B,H
27B3 4D       LD   C,L
27B4 29       ADD  HL,HL
27B5 D604     SUB  04H
27B7 3804     JR   C,27BDH
27B9 29       ADD  HL,HL
27BA 2806     JR   Z,27C2H
27BC 29       ADD  HL,HL
27BD B7       OR   A
27BE E2C227   JP   PO,27C2H
27C1 09       ADD  HL,BC
27C2 C1       POP  BC
27C3 09       ADD  HL,BC
27C4 EB       EX   DE,HL
27C5 2AF340   LD   HL,(40F3H)
27C8 C9       RET
27C9 AF       XOR  A
27CA E5       PUSH HL
27CB 32AF40   LD   (40AFH),A
27CE CDD427   CALL 27D4H
27D1 E1       POP  HL
27D2
27D3 C9       RET
27D4 2AFD40   LD   HL,(40FDH)
27D7 EB       EX   DE,HL
27D8 210000   LD   HL,0000H
27DB 39       ADD  HL,SP
27DC E7       RST  20H
27DD 200D     JR   NZ,27ECH
27DF CDDA29   CALL 29DAH
27E2 CDE628   CALL 28E6H
27E5 2AA040   LD   HL,(40A0H)
27E8 EB       EX   DE,HL
27E9 2AD640   LD   HL,(40D6H)
27EC 7D       LD   A,L
27ED 93       SUB  E
27EE 6F       LD   L,A
27EF 7C       LD   A,H
27F0 9A       SBC  A,D
27F1 67       LD   H,A
27F2 C3660C   JP   0C66H
27F5 3AA640   LD   A,(40A6H)
27F8 6F       LD   L,A
27F9 AF       XOR  A
27FA 67       LD   H,A
27FB C39A0A   JP   0A9AH
27FE CDA941   CALL 41A9H
2801
2802 CD2C25   CALL 252CH
2805 E5       PUSH HL
2806 219008   LD   HL,0890H
2809 E5       PUSH HL
280A 3AAF40   LD   A,(40AFH)
280D F5       PUSH AF
280E FE03     CP   03H
2810 CCDA29   CALL Z,29DAH
2813 F1       POP  AF
2814 EB       EX   DE,HL
2815 2A8E40   LD   HL,(408EH)
2818 E9       JP   (HL)
2819 E5       PUSH HL
281A E607     AND  07H
281C 21A118   LD   HL,18A1H
281F 4F       LD   C,A
2820 0600     LD   B,00H
2822 09       ADD  HL,BC
2823 CD8625   CALL 2586H
2826 E1       POP  HL
2827
2828 E5       PUSH HL
2829 2AA240   LD   HL,(40A2H)
282C 23       INC  HL
282D 7C       LD   A,H
282E B5       OR   L
282F E1       POP  HL
2830 C0       RET  NZ
2831 1E16     LD   E,16H
2833 C3A219   JP   19A2H
2836 CDBD0F   CALL 0FBDH
2839 CD6528   CALL 2865H
283C CDDA29   CALL 29DAH
283F 012B2A   LD   BC,2A2BH
2842 C5       PUSH BC
2843 7E       LD   A,(HL)
2844 23       INC  HL
2845 E5       PUSH HL
2846 CDBF28   CALL 28BFH
2849 E1       POP  HL
284A 4E       LD   C,(HL)
284B 23       INC  HL
284C 46       LD   B,(HL)
284D CD5A28   CALL 285AH
2850 E5       PUSH HL
2851 6F       LD   L,A
2852 CDCE29   CALL 29CEH
2855 D1       POP  DE
2856
2857 CDBF28   CALL 28BFH
285A 21D340   LD   HL,40D3H
285D E5       PUSH HL
285E 77       LD   (HL),A
285F 23       INC  HL
2860 73       LD   (HL),E
2861 23       INC  HL
2862 72       LD   (HL),D
2863 E1       POP  HL
2864 C9       RET
2865 2B       DEC  HL
2866 0622     LD   B,22H
2868 50       LD   D,B
2869 E5       PUSH HL
286A 0EFF     LD   C,0FFH
286C 23       INC  HL
```

```
286D 7E        LD   A,(HL)
286E 0C        INC  C
286F B7        OR   A
2870 2806      JR   Z,2878H
2872 BA        CP   D
2873 2803      JR   Z,2878H
2875 B8        CP   B
2876 20F4      JR   NZ,286CH
2878 FE22      CP   22H
287A CC781D    CALL Z,1D78H
287D E3        EX   (SP),HL
287E 23        INC  HL
287F EB        EX   DE,HL
2880 79        LD   A,C
2881 CD5A28    CALL 285AH
2884 11D340    LD   DE,40D3H
2887 3ED5      LD   A,0D5H
2889 2AB340    LD   HL,(40B3H)
288C 222141    LD   (4121H),HL
288F 3E03      LD   A,03H
2891 32AF40    LD   (40AFH),A
2894 CDD309    CALL 09D3H
2897 11D640    LD   DE,40D6H
289A DF        RST  18H
289B 22B340    LD   (40B3H),HL
289E E1        POP  HL
289F 7E        LD   A,(HL)
28A0
28A1 1E1E      LD   E,1EH
28A3 C3A219    JP   19A2H
28A6 23        INC  HL
28A7 CD6528    CALL 2865H
28AA CDDA29    CALL 29DAH
28AD CDC409    CALL 09C4H
28B0 14        INC  D
28B1 15        DEC  D
28B2
28B3 0A        LD   A,(BC)
28B4 CD2A03    CALL 032AH
28B7 FE0D      CP   0DH
28B9 CC0321    CALL Z,2103H
28BC 03        INC  BC
28BD 18F2      JR   28B1H
28BF B7        OR   A
28C0 0EF1      LD   C,0F1H
28C2 F5        PUSH AF
28C3 2AA040    LD   HL,(40A0H)
28C6 EB        EX   DE,HL
28C7 2AD640    LD   HL,(40D6H)
28CA 2F        CPL
28CB 4F        LD   C,A
28CC 06FF      LD   B,0FFH
28CE 09        ADD  HL,BC
28CF 23        INC  HL
28D0 DF        RST  18H
28D1 3807      JR   C,28DAH
28D3 22D640    LD   (40D6H),HL
28D6 23        INC  HL
28D7 EB        EX   DE,HL
28D8 F1        POP  AF
28D9
28DA F1        POP  AF
28DB 1E1A      LD   E,1AH
28DD CAA219    JP   Z,19A2H
28E0 BF        CP   A
28E1 F5        PUSH AF
28E2 01C128    LD   BC,28C1H
28E5 C5        PUSH BC
28E6 2AB140    LD   HL,(40B1H)
28E9 22D640    LD   (40D6H),HL
28EC 210000    LD   HL,0000H
28EF E5        PUSH HL
28F0 2AA040    LD   HL,(40A0H)
28F3 E5        PUSH HL
28F4 21B540    LD   HL,40B5H
28F7 EB        EX   DE,HL
28F8 2AB340    LD   HL,(40B3H)
28FB EB        EX   DE,HL
28FC DF        RST  18H
28FD 01F728    LD   BC,28F7H
2900 C24A29    JP   NZ,294AH
2903 2AF940    LD   HL,(40F9H)
2906 EB        EX   DE,HL
2907 2AFB40    LD   HL,(40FBH)
290A EB        EX   DE,HL
290B DF        RST  18H
290C 2813      JR   Z,2921H
290E 7E        LD   A,(HL)
290F 23        INC  HL
2910 23        INC  HL
2911 23        INC  HL
2912 FE03      CP   03H
2914 2004      JR   NZ,291AH
2916 CD4B29    CALL 294BH
2919 AF        XOR  A
291A 5F        LD   E,A
291B 1600      LD   D,00H
291D 19        ADD  HL,DE
291E 18E6      JR   2906H
2920 C1        POP  BC
2921 EB        EX   DE,HL
2922 2AFD40    LD   HL,(40FDH)
2925 EB        EX   DE,HL
2926
2927 CA6B29    JP   Z,296BH
292A 7E        LD   A,(HL)
292B 23        INC  HL
292C CDC209    CALL 09C2H
292F E5        PUSH HL
2930 09        ADD  HL,BC
2931 FE03      CP   03H
2933 20EB      JR   NZ,2920H
2935 22D840    LD   (40D8H),HL
2938 E1        POP  HL
2939 4E        LD   C,(HL)
293A 0600      LD   B,00H
293C 09        ADD  HL,BC
293D 09        ADD  HL,BC
293E 23        INC  HL
293F EB        EX   DE,HL
```

```
2940 2AD840    LD   HL,(40D8H)
2943 EB        EX   DE,HL
2944 DF        RST  18H
2945 28DA      JR   Z,2921H
2947 013F29    LD   BC,293FH
294A C5        PUSH BC
294B AF        XOR  A
294C B6        OR   (HL)
294D 23        INC  HL
294E 5E        LD   E,(HL)
294F 23        INC  HL
2950 56        LD   D,(HL)
2951 23        INC  HL
2952 C8        RET  Z
2953 44        LD   B,H
2954 4D        LD   C,L
2955 2AD640    LD   HL,(40D6H)
2958 DF        RST  18H
2959 60        LD   H,B
295A 69        LD   L,C
295B
295C E1        POP  HL
295D E3        EX   (SP),HL
295E DF        RST  18H
295F E3        EX   (SP),HL
2960 E5        PUSH HL
2961 60        LD   H,B
2962 69        LD   L,C
2963 D0        RET  NC
2964 C1        POP  BC
2965 F1        POP  AF
2966 F1        POP  AF
2967 E5        PUSH HL
2968 D5        PUSH DE
2969 C5        PUSH BC
296A C9        RET
296B D1        POP  DE
296C E1        POP  HL
296D 7D        LD   A,L
296E B4        OR   H
296F
2970 2B        DEC  HL
2971 46        LD   B,(HL)
2972 2B        DEC  HL
2973 4E        LD   C,(HL)
2974 E5        PUSH HL
2975 2B        DEC  HL
2976 6E        LD   L,(HL)
2977 2600      LD   H,00H
2979 09        ADD  HL,BC
297A 50        LD   D,B
297B 59        LD   E,C
297C 2B        DEC  HL
297D 44        LD   B,H
297E 4D        LD   C,L
297F 2AD640    LD   HL,(40D6H)
2982 CD5819    CALL 1958H
2985 E1        POP  HL
2986 71        LD   (HL),C
2987 23        INC  HL
2988 70        LD   (HL),B
2989 69        LD   L,C
298A 60        LD   H,B
298B 2B        DEC  HL
298C C3E928    JP   28E9H
298F C5        PUSH BC
2990 E5        PUSH HL
2991 2A2141    LD   HL,(4121H)
2994 E3        EX   (SP),HL
2995 CD9F24    CALL 249FH
2998 E3        EX   (SP),HL
2999 CDF40A    CALL 0AF4H
299C 7E        LD   A,(HL)
299D E5        PUSH HL
299E 2A2141    LD   HL,(4121H)
29A1 E5        PUSH HL
29A2 86        ADD  A,(HL)
29A3 1E1C      LD   E,1CH
29A5 DAA219    JP   C,19A2H
29A8 CD5728    CALL 2857H
29AB D1        POP  DE
29AC CDDE29    CALL 29DEH
29AF E3        EX   (SP),HL
29B0 CDDD29    CALL 29DDH
29B3 E5        PUSH HL
29B4 2AD440    LD   HL,(40D4H)
29B7 EB        EX   DE,HL
29B8 CDC629    CALL 29C6H
29BB CDC629    CALL 29C6H
29BE 214923    LD   HL,2349H
29C1 E3        EX   (SP),HL
29C2 E5        PUSH HL
29C3 C38428    JP   2884H
29C6 E1        POP  HL
29C7 E3        EX   (SP),HL
29C8 7E        LD   A,(HL)
29C9
29CA 4E        LD   C,(HL)
29CB
29CC 46        LD   B,(HL)
29CD 6F        LD   L,A
29CE 2C        INC  L
29CF 2D        DEC  L
29D0 C8        RET  Z
29D1 0A        LD   A,(BC)
29D2 12        LD   (DE),A
29D3 03        INC  BC
29D4 13        INC  DE
29D5 18F8      JR   29CFH
29D7 CDF40A    CALL 0AF4H
29DA 2A2141    LD   HL,(4121H)
29DD EB        EX   DE,HL
29DE CDF529    CALL 29F5H
29E1 EB        EX   DE,HL
29E2 C0        RET  NZ
29E3 D5        PUSH DE
29E4 50        LD   D,B
29E5 59        LD   E,C
29E6 1B        DEC  DE
29E7 4E        LD   C,(HL)
```

```
29E8 2AD640    LD   HL,(40D6H)
29EB DF        RST  18H
29EC 2005      JR   NZ,29F3H
29EE 47        LD   B,A
29EF 09        ADD  HL,BC
29F0 22D640    LD   (40D6H),HL
29F3 E1        POP  HL
29F4
29F5 2AB340    LD   HL,(40B3H)
29F8 2B        DEC  HL
29F9 46        LD   B,(HL)
29FA 2B        DEC  HL
29FB 4E        LD   C,(HL)
29FC 2B        DEC  HL
29FD DF        RST  18H
29FE C0        RET  NZ
29FF 22B340    LD   (40B3H),HL
2A02
2A03 01F827    LD   BC,27F8H
2A06 C5        PUSH BC
2A07 CDD729    CALL 29D7H
2A0A AF        XOR  A
2A0B 57        LD   D,A
2A0C 7E        LD   A,(HL)
2A0D B7        OR   A
2A0E C9        RET
2A0F 01F827    LD   BC,27F8H
2A12 C5        PUSH BC
2A13 CD072A    CALL 2A07H
2A16 CA4A1E    JP   Z,1E4AH
2A19 23        INC  HL
2A1A 5E        LD   E,(HL)
2A1B 23        INC  HL
2A1C 56        LD   D,(HL)
2A1D 1A        LD   A,(DE)
2A1E C9        RET
2A1F 3E01      LD   A,01H
2A21 CD5728    CALL 2857H
2A24 CD1F2B    CALL 2B1FH
2A27 2AD440    LD   HL,(40D4H)
2A2A 73        LD   (HL),E
2A2B C1        POP  BC
2A2C C38428    JP   2884H
2A2F D7        RST  10H
2A30 CF        RST  08H
2A31 28CD      JR   Z,2A00H
2A33 1C        INC  E
2A34 2B        DEC  HL
2A35 D5        PUSH DE
2A36 CF        RST  08H
2A37 2C        INC  L
2A38 CD3723    CALL 2337H
2A3B CF        RST  08H
2A3C 29        ADD  HL,HL
2A3D E3        EX   (SP),HL
2A3E E5        PUSH HL
2A3F E7        RST  20H
2A40 2805      JR   Z,2A47H
2A42 CD1F2B    CALL 2B1FH
2A45 1803      JR   2A4AH
2A47 CD132A    CALL 2A13H
2A4A D1        POP  DE
2A4B F5        PUSH AF
2A4C F5        PUSH AF
2A4D 7B        LD   A,E
2A4E CD5728    CALL 2857H
2A51 5F        LD   E,A
2A52 F1        POP  AF
2A53 1C        INC  E
2A54 1D        DEC  E
2A55 28D4      JR   Z,2A2BH
2A57 2AD440    LD   HL,(40D4H)
2A5A 77        LD   (HL),A
2A5B
2A5C 1D        DEC  E
2A5D 20FB      JR   NZ,2A5AH
2A5F 18CA      JR   2A2BH
2A61 CDDF2A    CALL 2ADFH
2A64 AF        XOR  A
2A65 E3        EX   (SP),HL
2A66 4F        LD   C,A
2A67 3EE5      LD   A,0E5H
2A69 E5        PUSH HL
2A6A 7E        LD   A,(HL)
2A6B B8        CP   B
2A6C 3802      JR   C,2A70H
2A6E 78        LD   A,B
2A6F 110E00    LD   DE,000EH
2A72 C5        PUSH BC
2A73 CDBF28    CALL 28BFH
2A76 C1        POP  BC
2A77 E1        POP  HL
2A78 E5        PUSH HL
2A79
2A7A 46        LD   B,(HL)
2A7B 23        INC  HL
2A7C 66        LD   H,(HL)
2A7D 68        LD   L,B
2A7E 0600      LD   B,00H
2A80 09        ADD  HL,BC
2A81 44        LD   B,H
2A82 4D        LD   C,L
2A83 CD5A28    CALL 285AH
2A86 6F        LD   L,A
2A87 CDCE29    CALL 29CEH
2A8A D1        POP  DE
2A8B CDDE29    CALL 29DEH
2A8E C38428    JP   2884H
2A91 CDDF2A    CALL 2ADFH
2A94 D1        POP  DE
2A95 D5        PUSH DE
2A96 1A        LD   A,(DE)
2A97 90        SUB  B
2A98 18CB      JR   2A65H
2A9A EB        EX   DE,HL
2A9B 7E        LD   A,(HL)
2A9C CDE22A    CALL 2AE2H
2A9F 04        INC  B
2AA0 05        DEC  B
2AA1 CA4A1E    JP   Z,1E4AH
```

```
2AA4 C5          PUSH BC
2AA5 1EFF        LD   E,0FFH
2AA7 FE29        CP   29H
2AA9 2805        JR   Z,2AB0H
2AAB CF          RST  08H
2AAC 2C          INC  L
2AAD CD1C2B      CALL 2B1CH
2AB0 CF          RST  08H
2AB1 29          ADD  HL,HL
2AB2 F1          POP  AF
2AB3 E3          EX   (SP),HL
2AB4 01692A      LD   BC,2A69H
2AB7 C5          PUSH BC
2AB8 3D          DEC  A
2AB9 BE          CP   (HL)
2ABA 0600        LD   B,00H
2ABC D0          RET  NC
2ABD 4F          LD   C,A
2ABE 7E          LD   A,(HL)
2ABF 91          SUB  C
2AC0 BB          CP   E
2AC1 47          LD   B,A
2AC2 D8          RET  C
2AC3 43          LD   B,E
2AC4
2AC5 CD072A      CALL 2A07H
2AC8 CAF827      JP   Z,27F8H
2ACB 5F          LD   E,A
2ACC 23          INC  HL
2ACD 7E          LD   A,(HL)
2ACE 23          INC  HL
2ACF 66          LD   H,(HL)
2AD0 6F          LD   L,A
2AD1 E5          PUSH HL
2AD2 19          ADD  HL,DE
2AD3 46          LD   B,(HL)
2AD4 72          LD   (HL),D
2AD5 E3          EX   (SP),HL
2AD6 C5          PUSH BC
2AD7 7E          LD   A,(HL)
2AD8 CD650E      CALL 0E65H
2ADB C1          POP  BC
2ADC E1          POP  HL
2ADD 70          LD   (HL),B
2ADE
2ADF EB          EX   DE,HL
2AE0 CF          RST  08H
2AE1 29          ADD  HL,HL
2AE2 C1          POP  BC
2AE3 D1          POP  DE
2AE4 C5          PUSH BC
2AE5 43          LD   B,E
2AE6 C9          RET
2AE7 FE7A        CP   7AH
2AE9 C29719      JP   NZ,1997H
2AEC C3D941      JP   41D9H
2AEF CD1F2B      CALL 2B1FH
2AF2 329440      LD   (4094H),A
2AF5 CD9340      CALL 4093H
2AF8 C3F827      JP   27F8H
2AFB CD0E2B      CALL 2B0EH
2AFE C39640      JP   4096H
2B01 D7          RST  10H
2B02 CD3723      CALL 2337H
2B05 E5          PUSH HL
2B06 CD7F0A      CALL 0A7FH
2B09 EB          EX   DE,HL
2B0A E1          POP  HL
2B0B 7A          LD   A,D
2B0C B7          OR   A
2B0D
2B0E CD1C2B      CALL 2B1CH
2B11 329440      LD   (4094H),A
2B14 329740      LD   (4097H),A
2B17 CF          RST  08H
2B18 2C          INC  L
2B19 1801        JR   2B1CH
2B1B D7          RST  10H
2B1C CD3723      CALL 2337H
2B1F CD052B      CALL 2B05H
2B22 C24A1E      JP   NZ,1E4AH
2B25 2B          DEC  HL
2B26 D7          RST  10H
2B27 7B          LD   A,E
2B28
2B29 3E01        LD   A,01H
2B2B 329C40      LD   (409CH),A
2B2E C1          POP  BC
2B2F CD101B      CALL 1B10H
2B32 C5          PUSH BC
2B33 21FFFF      LD   HL,0FFFFH
2B36 22A240      LD   (40A2H),HL
2B39 E1          POP  HL
2B3A D1          POP  DE
2B3B 4E          LD   C,(HL)
2B3C 23          INC  HL
2B3D 46          LD   B,(HL)
2B3E 23          INC  HL
2B3F 78          LD   A,B
2B40 B1          OR   C
2B41 CA191A      JP   Z,1A19H
2B44 CDDF41      CALL 41DFH
2B47 CD9B1D      CALL 1D9BH
2B4A C5          PUSH BC
2B4B 4E          LD   C,(HL)
2B4C 23          INC  HL
2B4D 46          LD   B,(HL)
2B4E 23          INC  HL
2B4F C5          PUSH BC
2B50 E3          EX   (SP),HL
2B51 EB          EX   DE,HL
2B52 DF          RST  18H
2B53 C1          POP  BC
2B54 DA181A      JP   C,1A18H
2B57 E3          EX   (SP),HL
2B58 E5          PUSH HL
2B59 C5          PUSH BC
2B5A EB          EX   DE,HL
2B5B 22EC40      LD   (40ECH),HL
2B5E CDAF0F      CALL 0FAFH
```

```
2B61 3E20      LD   A,20H
2B63 E1        POP  HL
2B64 CD2A03    CALL 032AH
2B67 CD7E2B    CALL 2B7EH
2B6A 2AA740    LD   HL,(40A7H)
2B6D CD752B    CALL 2B75H
2B70 CDFE20    CALL 20FEH
2B73 18BE      JR   2B33H
2B75 7E        LD   A,(HL)
2B76 B7        OR   A
2B77 C8        RET  Z
2B78 CD2A03    CALL 032AH
2B7B 23        INC  HL
2B7C 18F7      JR   2B75H
2B7E E5        PUSH HL
2B7F 2AA740    LD   HL,(40A7H)
2B82 44        LD   B,H
2B83 4D        LD   C,L
2B84 E1        POP  HL
2B85 16FF      LD   D,0FFH
2B87 1803      JR   2B8CH
2B89 03        INC  BC
2B8A
2B8B C8        RET  Z
2B8C 7E        LD   A,(HL)
2B8D B7        OR   A
2B8E 23        INC  HL
2B8F 02        LD   (BC),A
2B90 C8        RET  Z
2B91 F2892B    JP   P,2B89H
2B94 FEFB      CP   0FBH
2B96 2008      JR   NZ,2BA0H
2B98 0B        DEC  BC
2B99 0B        DEC  BC
2B9A 0B        DEC  BC
2B9B 0B        DEC  BC
2B9C 14        INC  D
2B9D 14        INC  D
2B9E 14        INC  D
2B9F 14        INC  D
2BA0 FE95      CP   95H
2BA2 CC240B    CALL Z,0B24H
2BA5 D67F      SUB  7FH
2BA7 E5        PUSH HL
2BA8 5F        LD   E,A
2BA9 215016    LD   HL,1650H
2BAC 7E        LD   A,(HL)
2BAD B7        OR   A
2BAE 23        INC  HL
2BAF F2AC2B    JP   P,2BACH
2BB2 1D        DEC  E
2BB3 20F7      JR   NZ,2BACH
2BB5 E67F      AND  7FH
2BB7 02        LD   (BC),A
2BB8 03        INC  BC
2BB9
2BBA CAD828    JP   Z,28D8H
2BBD 7E        LD   A,(HL)
2BBE 23        INC  HL
2BBF B7        OR   A
2BC0 F2B72B    JP   P,2BB7H
2BC3 E1        POP  HL
2BC4 18C6      JR   2B8CH
2BC6 CD101B    CALL 1B10H
2BC9 D1        POP  DE
2BCA C5        PUSH BC
2BCB C5        PUSH BC
2BCC CD2C1B    CALL 1B2CH
2BCF 3005      JR   NC,2BD6H
2BD1 54        LD   D,H
2BD2 5D        LD   E,L
2BD3 E3        EX   (SP),HL
2BD4 E5        PUSH HL
2BD5
2BD6 D24A1E    JP   NC,1E4AH
2BD9 212919    LD   HL,1929H
2BDC CDA728    CALL 28A7H
2BDF C1        POP  BC
2BE0 21E81A    LD   HL,1AE8H
2BE3 E3        EX   (SP),HL
2BE4 EB        EX   DE,HL
2BE5 2AF940    LD   HL,(40F9H)
2BE8 1A        LD   A,(DE)
2BE9 02        LD   (BC),A
2BEA 03        INC  BC
2BEB 13        INC  DE
2BEC DF        RST  18H
2BED 20F9      JR   NZ,2BE8H
2BEF 60        LD   H,B
2BF0 69        LD   L,C
2BF1 22F940    LD   (40F9H),HL
2BF4 C9        RET
2BF5 CD8402    CALL 0284H
2BF8 CD3723    CALL 2337H
2BFB E5        PUSH HL
2BFC CD132A    CALL 2A13H
2BFF 3ED3      LD   A,0D3H
2C01 CD6402    CALL 0264H
2C04 CD6102    CALL 0261H
2C07 1A        LD   A,(DE)
2C08 CD6402    CALL 0264H
2C0B 2AA440    LD   HL,(40A4H)
2C0E EB        EX   DE,HL
2C0F 2AF940    LD   HL,(40F9H)
2C12 1A        LD   A,(DE)
2C13 13        INC  DE
2C14 CD6402    CALL 0264H
2C17 DF        RST  18H
2C18 20F8      JR   NZ,2C12H
2C1A CDF801    CALL 01F8H
2C1D E1        POP  HL
2C1E
2C1F CD9302    CALL 0293H
2C22 7E        LD   A,(HL)
2C23 D6B2      SUB  0B2H
2C25 2802      JR   Z,2C29H
2C27 AF        XOR  A
2C28 012F23    LD   BC,232FH
2C2B F5        PUSH AF
2C2C 2B        DEC  HL
```

```
2C2D D7        RST  10H
2C2E 3E00      LD   A,00H
2C30 2807      JR   Z,2C39H
2C32 CD3723    CALL 2337H
2C35 CD132A    CALL 2A13H
2C38 1A        LD   A,(DE)
2C39 6F        LD   L,A
2C3A F1        POP  AF
2C3B
2C3C 67        LD   H,A
2C3D 222141    LD   (4121H),HL
2C40 CC4D1B    CALL Z,1B4DH
2C43 2A2141    LD   HL,(4121H)
2C46 EB        EX   DE,HL
2C47 0603      LD   B,03H
2C49 CD3502    CALL 0235H
2C4C D6D3      SUB  0D3H
2C4E 20F7      JR   NZ,2C47H
2C50 10F7      DJNZ 2C49H
2C52 CD3502    CALL 0235H
2C55 1C        INC  E
2C56 1D        DEC  E
2C57 2803      JR   Z,2C5CH
2C59 BB        CP   E
2C5A 2037      JR   NZ,2C93H
2C5C 2AA440    LD   HL,(40A4H)
2C5F 0603      LD   B,03H
2C61 CD3502    CALL 0235H
2C64 5F        LD   E,A
2C65 96        SUB  (HL)
2C66 A2        AND  D
2C67 2021      JR   NZ,2C8AH
2C69 73        LD   (HL),E
2C6A CD6C19    CALL 196CH
2C6D 7E        LD   A,(HL)
2C6E B7        OR   A
2C6F 23        INC  HL
2C70 20ED      JR   NZ,2C5FH
2C72 CD2C02    CALL 022CH
2C75 10EA      DJNZ 2C61H
2C77 22F940    LD   (40F9H),HL
2C7A 212919    LD   HL,1929H
2C7D CDA728    CALL 28A7H
2C80 CDF801    CALL 01F8H
2C83 2AA440    LD   HL,(40A4H)
2C86 E5        PUSH HL
2C87 C3E81A    JP   1AE8H
2C8A 21A52C    LD   HL,2CA5H
2C8D CDA728    CALL 28A7H
2C90 C3181A    JP   1A18H
2C93 323E3C    LD   (3C3EH),A
2C96 0603      LD   B,03H
2C98 CD3502    CALL 0235H
2C9B
2C9C 20F8      JR   NZ,2C96H
2C9E 10F8      DJNZ 2C98H
2CA0 CD9602    CALL 0296H
2CA3 18A2      JR   2C47H
2CA5 42        LD   B,D
2CA6 41        LD   B,C
2CA7 44        LD   B,H
2CA8 0D        DEC  C
2CA9 00        NOP
2CAA CD7F0A    CALL 0A7FH
2CAD 7E        LD   A,(HL)
2CAE C3F827    JP   27F8H
2CB1 CD022B    CALL 2B02H
2CB4 D5        PUSH DE
2CB5 CF        RST  08H
2CB6 2C        INC  L
2CB7 CD1C2B    CALL 2B1CH
2CBA D1        POP  DE
2CBB 12        LD   (DE),A
2CBC
2CBD CD3823    CALL 2338H
2CC0 CDF40A    CALL 0AF4H
2CC3 CF        RST  08H
2CC4 3B        DEC  SP
2CC5 EB        EX   DE,HL
2CC6 2A2141    LD   HL,(4121H)
2CC9 1808      JR   2CD3H
2CCB 3ADE40    LD   A,(40DEH)
2CCE B7        OR   A
2CCF 280C      JR   Z,2CDDH
2CD1 D1        POP  DE
2CD2 EB        EX   DE,HL
2CD3 E5        PUSH HL
2CD4 AF        XOR  A
2CD5 32DE40    LD   (40DEH),A
2CD8 BA        CP   D
2CD9 F5        PUSH AF
2CDA D5        PUSH DE
2CDB 46        LD   B,(HL)
2CDC B0        OR   B
2CDD CA4A1E    JP   Z,1E4AH
2CE0 23        INC  HL
2CE1 4E        LD   C,(HL)
2CE2 23        INC  HL
2CE3 66        LD   H,(HL)
2CE4 69        LD   L,C
2CE5 181C      JR   2D03H
2CE7 58        LD   E,B
2CE8 E5        PUSH HL
2CE9 0E02      LD   C,02H
2CEB 7E        LD   A,(HL)
2CEC 23        INC  HL
2CED FE25      CP   25H
2CEF CA172E    JP   Z,2E17H
2CF2 FE20      CP   20H
2CF4 2003      JR   NZ,2CF9H
2CF6 0C        INC  C
2CF7 10F2      DJNZ 2CEBH
2CF9 E1        POP  HL
2CFA 43        LD   B,E
2CFB 3E25      LD   A,25H
2CFD CD492E    CALL 2E49H
2D00 CD2A03    CALL 032AH
2D03 AF        XOR  A
2D04 5F        LD   E,A
2D05 57        LD   D,A
```

```
2D06 CD492E     CALL 2E49H
2D09 57         LD   D,A
2D0A 7E         LD   A,(HL)
2D0B 23         INC  HL
2D0C FE21       CP   21H
2D0E CA142E     JP   Z,2E14H
2D11 FE23       CP   23H
2D13 2837       JR   Z,2D4CH
2D15 05         DEC  B
2D16 CAFE2D     JP   Z,2DFEH
2D19 FE2B       CP   2BH
2D1B 3E08       LD   A,08H
2D1D 28E7       JR   Z,2D06H
2D1F 2B         DEC  HL
2D20 7E         LD   A,(HL)
2D21
2D22 FE2E       CP   2EH
2D24 2840       JR   Z,2D66H
2D26 FE25       CP   25H
2D28 28BD       JR   Z,2CE7H
2D2A BE         CP   (HL)
2D2B 20D0       JR   NZ,2CFDH
2D2D FE24       CP   24H
2D2F 2814       JR   Z,2D45H
2D31 FE2A       CP   2AH
2D33 20C8       JR   NZ,2CFDH
2D35 78         LD   A,B
2D36 FE02       CP   02H
2D38 23         INC  HL
2D39 3803       JR   C,2D3EH
2D3B 7E         LD   A,(HL)
2D3C FE24       CP   24H
2D3E 3E20       LD   A,20H
2D40 2007       JR   NZ,2D49H
2D42 05         DEC  B
2D43 1C         INC  E
2D44 FEAF       CP   0AFH
2D46 C610       ADD  10H
2D48 23         INC  HL
2D49 1C         INC  E
2D4A 82         ADD  A,D
2D4B 57         LD   D,A
2D4C 1C         INC  E
2D4D 0E00       LD   C,00H
2D4F 05         DEC  B
2D50 2847       JR   Z,2D99H
2D52 7E         LD   A,(HL)
2D53
2D54 FE2E       CP   2EH
2D56 2818       JR   Z,2D70H
2D58 FE23       CP   23H
2D5A 28F0       JR   Z,2D4CH
2D5C FE2C       CP   2CH
2D5E 201A       JR   NZ,2D7AH
2D60 7A         LD   A,D
2D61 F640       OR   40H
2D63 57         LD   D,A
2D64 18E6       JR   2D4CH
2D66 7E         LD   A,(HL)
2D67 FE23       CP   23H
2D69 3E2E       LD   A,2EH
2D6B 2090       JR   NZ,2CFDH
2D6D 0E01       LD   C,01H
2D6F 23         INC  HL
2D70 0C         INC  C
2D71 05         DEC  B
2D72 2825       JR   Z,2D99H
2D74 7E         LD   A,(HL)
2D75 23         INC  HL
2D76 FE23       CP   23H
2D78 28F6       JR   Z,2D70H
2D7A D5         PUSH DE
2D7B 11972D     LD   DE,2D97H
2D7E D5         PUSH DE
2D7F 54         LD   D,H
2D80 5D         LD   E,L
2D81 FE5B       CP   5BH
2D83 C0         RET  NZ
2D84 BE         CP   (HL)
2D85
2D86 23         INC  HL
2D87 BE         CP   (HL)
2D88 C0         RET  NZ
2D89 23         INC  HL
2D8A BE         CP   (HL)
2D8B C0         RET  NZ
2D8C 23         INC  HL
2D8D 78         LD   A,B
2D8E D604       SUB  04H
2D90
2D91 D1         POP  DE
2D92 D1         POP  DE
2D93 47         LD   B,A
2D94 14         INC  D
2D95 23         INC  HL
2D96 CAEBD1     JP   Z,0D1EBH
2D99 7A         LD   A,D
2D9A 2B         DEC  HL
2D9B 1C         INC  E
2D9C E608       AND  08H
2D9E 2015       JR   NZ,2DB5H
2DA0 1D         DEC  E
2DA1 78         LD   A,B
2DA2 B7         OR   A
2DA3 2810       JR   Z,2DB5H
2DA5 7E         LD   A,(HL)
2DA6 D62D       SUB  2DH
2DA8 2806       JR   Z,2DB0H
2DAA FEFE       CP   0FEH
2DAC 2007       JR   NZ,2DB5H
2DAE 3E08       LD   A,08H
2DB0 C604       ADD  04H
2DB2 82         ADD  A,D
2DB3 57         LD   D,A
2DB4 05         DEC  B
2DB5 E1         POP  HL
2DB6 F1         POP  AF
2DB7 2850       JR   Z,2E09H
2DB9 C5         PUSH BC
2DBA D5         PUSH DE
```

```
2DBB CD3723     CALL 2337H
2DBE D1         POP  DE
2DBF C1         POP  BC
2DC0 C5         PUSH BC
2DC1 E5         PUSH HL
2DC2 43         LD   B,E
2DC3 78         LD   A,B
2DC4 81         ADD  A,C
2DC5 FE19       CP   19H
2DC7 D24A1E     JP   NC,1E4AH
2DCA 7A         LD   A,D
2DCB F680       OR   80H
2DCD CDBE0F     CALL 0FBEH
2DD0 CDA728     CALL 28A7H
2DD3 E1         POP  HL
2DD4 2B         DEC  HL
2DD5 D7         RST  10H
2DD6 37         SCF
2DD7 280D       JR   Z,2DE6H
2DD9 32DE40     LD   (40DEH),A
2DDC FE3B       CP   3BH
2DDE 2805       JR   Z,2DE5H
2DE0 FE2C       CP   2CH
2DE2 C29719     JP   NZ,1997H
2DE5 D7         RST  10H
2DE6 C1         POP  BC
2DE7
2DE8 E1         POP  HL
2DE9 E5         PUSH HL
2DEA F5         PUSH AF
2DEB D5         PUSH DE
2DEC 7E         LD   A,(HL)
2DED 90         SUB  B
2DEE 23         INC  HL
2DEF 4E         LD   C,(HL)
2DF0 23         INC  HL
2DF1 66         LD   H,(HL)
2DF2 69         LD   L,C
2DF3 1600       LD   D,00H
2DF5 5F         LD   E,A
2DF6 19         ADD  HL,DE
2DF7 78         LD   A,B
2DF8 B7         OR   A
2DF9 C2032D     JP   NZ,2D03H
2DFC 1806       JR   2E04H
2DFE CD492E     CALL 2E49H
2E01 CD2A03     CALL 032AH
2E04 E1         POP  HL
2E05 F1         POP  AF
2E06 C2CB2C     JP   NZ,2CCBH
2E09 DCFE20     CALL C,20FEH
2E0C
2E0D CDDD29     CALL 29DDH
2E10 E1         POP  HL
2E11 C36921     JP   2169H
2E14 0E01       LD   C,01H
2E16 3EF1       LD   A,0F1H
2E18 05         DEC  B
2E19 CD492E     CALL 2E49H
2E1C E1         POP  HL
2E1D F1         POP  AF
2E1E 28E9       JR   Z,2E09H
2E20 C5         PUSH BC
2E21 CD3723     CALL 2337H
2E24 CDF40A     CALL 0AF4H
2E27 C1         POP  BC
2E28 C5         PUSH BC
2E29 E5         PUSH HL
2E2A 2A2141     LD   HL,(4121H)
2E2D 41         LD   B,C
2E2E 0E00       LD   C,00H
2E30 C5         PUSH BC
2E31 CD682A     CALL 2A68H
2E34 CDAA28     CALL 28AAH
2E37 2A2141     LD   HL,(4121H)
2E3A F1         POP  AF
2E3B 96         SUB  (HL)
2E3C 47         LD   B,A
2E3D 3E20       LD   A,20H
2E3F 04         INC  B
2E40 05         DEC  B
2E41 CAD32D     JP   Z,2DD3H
2E44 CD2A03     CALL 032AH
2E47 18F7       JR   2E40H
2E49 F5         PUSH AF
2E4A 7A         LD   A,D
2E4B B7         OR   A
2E4C 3E2B       LD   A,2BH
2E4E C42A03     CALL NZ,032AH
2E51 F1         POP  AF
2E52
2E53 329A40     LD   (409AH),A
2E56 2AEA40     LD   HL,(40EAH)
2E59 B4         OR   H
2E5A A5         AND  L
2E5B 3C         INC  A
2E5C EB         EX   DE,HL
2E5D C8         RET  Z
2E5E 1804       JR   2E64H
2E60 CD4F1E     CALL 1E4FH
2E63
2E64 E1         POP  HL
2E65 EB         EX   DE,HL
2E66 22EC40     LD   (40ECH),HL
2E69 EB         EX   DE,HL
2E6A CD2C1B     CALL 1B2CH
2E6D D2D91E     JP   NC,1ED9H
2E70 60         LD   H,B
2E71 69         LD   L,C
2E72 23         INC  HL
2E73 23         INC  HL
2E74 4E         LD   C,(HL)
2E75 23         INC  HL
2E76 46         LD   B,(HL)
2E77 23         INC  HL
2E78 C5         PUSH BC
2E79 CD7E2B     CALL 2B7EH
2E7C E1         POP  HL
2E7D E5         PUSH HL
2E7E CDAF0F     CALL 0FAFH
```

```
2E81 3E20      LD   A,20H
2E83 CD2A03    CALL 032AH
2E86 2AA740    LD   HL,(40A7H)
2E89 3E0E      LD   A,0EH
2E8B CD2A03    CALL 032AH
2E8E E5        PUSH HL
2E8F 0EFF      LD   C,0FFH
2E91 0C        INC  C
2E92 7E        LD   A,(HL)
2E93 B7        OR   A
2E94 23        INC  HL
2E95 20FA      JR   NZ,2E91H
2E97 E1        POP  HL
2E98 47        LD   B,A
2E99 1600      LD   D,00H
2E9B CD8403    CALL 0384H
2E9E D630      SUB  30H
2EA0 380E      JR   C,2EB0H
2EA2 FE0A      CP   0AH
2EA4 300A      JR   NC,2EB0H
2EA6 5F        LD   E,A
2EA7 7A        LD   A,D
2EA8
2EA9
2EAA 82        ADD  A,D
2EAB 07        RLCA
2EAC 83        ADD  A,E
2EAD 57        LD   D,A
2EAE 18EB      JR   2E9BH
2EB0 E5        PUSH HL
2EB1 21992E    LD   HL,2E99H
2EB4 E3        EX   (SP),HL
2EB5 15        DEC  D
2EB6 14        INC  D
2EB7 C2BB2E    JP   NZ,2EBBH
2EBA 14        INC  D
2EBB FED8      CP   0D8H
2EBD CAD22F    JP   Z,2FD2H
2EC0 FEDD      CP   0DDH
2EC2 CAE02F    JP   Z,2FE0H
2EC5 FEF0      CP   0F0H
2EC7 2841      JR   Z,2F0AH
2EC9 FE31      CP   31H
2ECB 3802      JR   C,2ECFH
2ECD D620      SUB  20H
2ECF FE21      CP   21H
2ED1 CAF62F    JP   Z,2FF6H
2ED4 FE1C      CP   1CH
2ED6 CA402F    JP   Z,2F40H
2ED9 FE23      CP   23H
2EDB 283F      JR   Z,2F1CH
2EDD FE19      CP   19H
2EDF CA7D2F    JP   Z,2F7DH
2EE2 FE14      CP   14H
2EE4 CA4A2F    JP   Z,2F4AH
2EE7 FE13      CP   13H
2EE9 CA652F    JP   Z,2F65H
2EEC FE15      CP   15H
2EEE CAE32F    JP   Z,2FE3H
2EF1 FE28      CP   28H
2EF3 CA782F    JP   Z,2F78H
2EF6 FE1B      CP   1BH
2EF8 281C      JR   Z,2F16H
2EFA FE18      CP   18H
2EFC CA752F    JP   Z,2F75H
2EFF FE11      CP   11H
2F01 C0        RET  NZ
2F02 C1        POP  BC
2F03 D1        POP  DE
2F04 CDFE20    CALL 20FEH
2F07 C3652E    JP   2E65H
2F0A 7E        LD   A,(HL)
2F0B B7        OR   A
2F0C C8        RET  Z
2F0D 04        INC  B
2F0E CD2A03    CALL 032AH
2F11 23        INC  HL
2F12 15        DEC  D
2F13 20F5      JR   NZ,2F0AH
2F15
2F16 E5        PUSH HL
2F17 215F2F    LD   HL,2F5FH
2F1A E3        EX   (SP),HL
2F1B 37        SCF
2F1C F5        PUSH AF
2F1D CD8403    CALL 0384H
2F20 5F        LD   E,A
2F21 F1        POP  AF
2F22 F5        PUSH AF
2F23 DC5F2F    CALL C,2F5FH
2F26 7E        LD   A,(HL)
2F27 B7        OR   A
2F28 CA3E2F    JP   Z,2F3EH
2F2B CD2A03    CALL 032AH
2F2E F1        POP  AF
2F2F F5        PUSH AF
2F30 DCA12F    CALL C,2FA1H
2F33 3802      JR   C,2F37H
2F35 23        INC  HL
2F36 04        INC  B
2F37 7E        LD   A,(HL)
2F38 BB        CP   E
2F39 20EB      JR   NZ,2F26H
2F3B 15        DEC  D
2F3C 20E8      JR   NZ,2F26H
2F3E F1        POP  AF
2F3F
2F40 CD752B    CALL 2B75H
2F43 CDFE20    CALL 20FEH
2F46 C1        POP  BC
2F47 C37C2E    JP   2E7CH
2F4A 7E        LD   A,(HL)
2F4B B7        OR   A
2F4C C8        RET  Z
2F4D 3E21      LD   A,21H
2F4F CD2A03    CALL 032AH
2F52 7E        LD   A,(HL)
2F53 B7        OR   A
2F54 2809      JR   Z,2F5FH
2F56 CD2A03    CALL 032AH
```

```
2F59 CDA12F     CALL 2FA1H
2F5C 15         DEC  D
2F5D 20F3       JR   NZ,2F52H
2F5F 3E21       LD   A,21H
2F61 CD2A03     CALL 032AH
2F64
2F65 7E         LD   A,(HL)
2F66 B7         OR   A
2F67 C8         RET  Z
2F68 CD8403     CALL 0384H
2F6B 77         LD   (HL),A
2F6C CD2A03     CALL 032AH
2F6F 23         INC  HL
2F70 04         INC  B
2F71 15         DEC  D
2F72 20F1       JR   NZ,2F65H
2F74
2F75 3600       LD   (HL),00H
2F77 48         LD   C,B
2F78 16FF       LD   D,0FFH
2F7A CD0A2F     CALL 2F0AH
2F7D CD8403     CALL 0384H
2F80 B7         OR   A
2F81 CA7D2F     JP   Z,2F7DH
2F84 FE08       CP   08H
2F86 280A       JR   Z,2F92H
2F88 FE0D       CP   0DH
2F8A CAE02F     JP   Z,2FE0H
2F8D FE1B       CP   1BH
2F8F C8         RET  Z
2F90 201E       JR   NZ,2FB0H
2F92 3E08       LD   A,08H
2F94 05         DEC  B
2F95 04         INC  B
2F96 281F       JR   Z,2FB7H
2F98 CD2A03     CALL 032AH
2F9B 2B         DEC  HL
2F9C 05         DEC  B
2F9D 117D2F     LD   DE,2F7DH
2FA0 D5         PUSH DE
2FA1 E5         PUSH HL
2FA2 0D         DEC  C
2FA3 7E         LD   A,(HL)
2FA4 B7         OR   A
2FA5 37         SCF
2FA6 CA9008     JP   Z,0890H
2FA9 23         INC  HL
2FAA 7E         LD   A,(HL)
2FAB 2B         DEC  HL
2FAC 77         LD   (HL),A
2FAD 23         INC  HL
2FAE 18F3       JR   2FA3H
2FB0 F5         PUSH AF
2FB1 79         LD   A,C
2FB2 FEFF       CP   0FFH
2FB4 3803       JR   C,2FB9H
2FB6 F1         POP  AF
2FB7 18C4       JR   2F7DH
2FB9 90         SUB  B
2FBA 0C         INC  C
2FBB 04         INC  B
2FBC C5         PUSH BC
2FBD
2FBE 6F         LD   L,A
2FBF 2600       LD   H,00H
2FC1 19         ADD  HL,DE
2FC2 44         LD   B,H
2FC3 4D         LD   C,L
2FC4 23         INC  HL
2FC5 CD5819     CALL 1958H
2FC8 C1         POP  BC
2FC9 F1         POP  AF
2FCA 77         LD   (HL),A
2FCB CD2A03     CALL 032AH
2FCE 23         INC  HL
2FCF C37D2F     JP   2F7DH
2FD2 78         LD   A,B
2FD3 B7         OR   A
2FD4 C8         RET  Z
2FD5 05         DEC  B
2FD6 2B         DEC  HL
2FD7 3E08       LD   A,08H
2FD9 CD2A03     CALL 032AH
2FDC 15         DEC  D
2FDD 20F3       JR   NZ,2FD2H
2FDF
2FE0 CD752B     CALL 2B75H
2FE3 CDFE20     CALL 20FEH
2FE6 C1         POP  BC
2FE7 D1         POP  DE
2FE8 7A         LD   A,D
2FE9 A3         AND  E
2FEA 3C         INC  A
2FEB 2AA740     LD   HL,(40A7H)
2FEE 2B         DEC  HL
2FEF C8         RET  Z
2FF0 37         SCF
2FF1 23         INC  HL
2FF2 F5         PUSH AF
2FF3 C3981A     JP   1A98H
2FF6 C1         POP  BC
2FF7 D1         POP  DE
2FF8 C3191A     JP   1A19H
2FFB DEC3       SBC  0C3H
2FFD C344B2     JP   0B244H
3000 FF         RST  38H
```

- CHAPTER 3 -

USING VARPTR AND ROM CALLS FOR INTEGERS

INTRODUCTION:

The VARPTR function in Level II BASIC and its equivalent CALL in Level II ROM is one of the most useful, but unfortunately neglected functions available to the TRS-80 programmer. A search through the literature available on the TRS-80 strongly supports this conclusion. Why the benign neglect of this very useful function? We will make three guesses:

1. A number of writers do not truly understand its function.

2. Those that understand its function do not fully understand how to use it.

3. Those that understand how to use it are intimidated and/or overwhelmed by the supposed complexities of the normalized exponential format with which single and double precision numbers are stored in MEM by Level II BASIC, the ACCUM, and "CS" = CDBL Store.

Every supposedly complex gizmo, whether it is an algorithm, a mathematical formula, a law of nature, or even a complicated computer program is in reality nothing more than a 'whole bunch' of simple and easily understood facts of life/nature grouped together. Taken as the whole they may seem awesome indeed, but when logically disassembled into their minimum divisible component parts, they may usually be readily understood by the average 8th grader.

In this and the next few Chapters let us massage and manipulate VARPTR and its associated equivalent CALL subroutines to the point where we all understand the subject thoroughly. By thoroughly we mean that YOU can explain it in relatively simple terms to a newcomer to the subject.

We will start with simple integers, grope upwards and onwards into strings and string/arrays, and then tackle normalized exponential formats for both single and double precision floating point numbers. Henceforth, anytime we mention single and double precision numbers, we will presume that you know we mean single and/or double precision 'floating point' numbers. The utter simplicity and sheer logic is indeed a pleasure to behold. We will use the same format as Volume 1 of the Disassembled Handbook and approach our enemy from the rear by using Level II BASIC for starters, and progress into assembly language to totally encircle and capture the subject.

So much for philosophizing for the time being. "The 'Bandits at 3 o'clock high,' may indeed turn out to tame ducks of the first water," the Presidential candidate said. Now let's get on with the subject at hand.

VARPTR BASIC REVIEW of INTEGERS:

The Level II BASIC Reference Manual's Chapter 8 coverage of the VARPTR function is abysmally lacking in sum and substance. The manuals published through about spring '79 were totally WRONG in the EXAMPLES paragraph for single and double precision, lines 3, 4, and 9 which undoubtedly has driven many TRS-80 buffs up the wall, plus they did not give any examples of the "two's complement form," that was mentioned in the integer variable paragraph. Why? Our best guesstimate is that the Radio Shack personnel who wrote the manual did not have the slightest idea or comprehension of what they were discussing. They were just repeating by rote what Microsoft had told them over a VERY noisy phone line, and only part of the message got through. Sad to say, the noise was printed as part of the manual. Now let's separate the fact from fiction.

INTEGER VARIABLES:

An integer variable may contain any round number value between -32768 and +32767. Do not confuse DEFINT variable with INT (x) as they have little in common. DEFINT is limited to the range -32768 to +32767 while INT (x) performs the rounding to a whole number of ANY number out to 17 digits.

Why the -32768 to +32767 range for integer variables? If you recall from Volume 1 of the Disassembled Handbook, all integer arithmetic is performed using the DE and HL register pairs, plus 2 spots in the ACCUM at RAM MEM locations 411DH through 4124H. The largest binary number our 8 bit Z-80 can hold in registers D or H is binary 11111111 and in registers E or L is binary 11111111. Hence the maximum number one register PAIR can hold is decimal 65535 = binary 1111111111111111. Try adding 32768 and 32767 together and we get 65535. Oho, there is a certain method to this madness. Microsoft and the other genii at Zilog very thoughtfully accomodated our need to work with negative as well as positive numbers by splitting the total integers available to us nearly in half. Remember, that integer variables buy us one nearly priceless commodity; i.e., speed, when compared to single or double precision arithmetic processing time.

Ok, let's try our VARiable PoinTeR function on a simple 'one liner' program in BASIC using an integer. Type in this program and RUN:

```
10 CLS:DEFINTA:A=10000:L=VARPTR(A):PRINTL;L+1
```

If you are in BASIC2 of disk or just plain 'ole non-disk Level II BASIC you should receive an output of:

```
17165      17166      (or thereabouts)
```

The first MEM location contains the LSB (Least Significant Byte) and the second MEM location the MSB (Most Significant Byte) of our integer = 10000, a nice round number.

Let's add another line to our program to PEEK the values stored in MEM for our integer and then RUN.

```
10 CLS:DEFINTA:A=10000:L=VARPTR(A):PRINTL,L+1
20 PRINTPEEK(L),PEEK(L+1)
```

```
OUTPUT:     17182          17183
            16             39
```

Our integer storage locations moved UPWARDS about 17 MEM locations. A. Why? B. Is it important?

A. Line 20's instructions took the extra 17 bytes.

B. Of no significance whatsoever. Just remember that the 2 byte storage location in MEM of an integer is usually changed when extra program bytes are added AFTER the integer statement (line 10).

JP back to Volume 1's Multi-Base Number Conversion Program and use the "SPLIT DECIMAL TO DECIMAL ENTER SP" to obtain the number represented by 16 and 39, above. Amazing, it = 10000. Is there an easier way to convert 16 and 39? There sure is. Type in the following additional line and then RUN.

```
30 B=PEEK(L)+256*PEEK(L+1):PRINTB
```

```
OUTPUT:     17207          17208
            16             39
            10000
```

Fantastic. This simple equation converts TRS-80 SPLIT DECIMAL TO DECIMAL and prints it out with only a single line. Why did not the author use it in Volume 1?

ANSWER: For three reasons.

1. It does not print out the hex number first.
2. Chapter 1's conversion program was already written and had the decimal to hex and hex to decimal conversion routines in it. The only modification necessary to make it work with SPLIT DECIMAL was to reverse the 2 equivalent hex numbers.
3. We are inherently very lazy.

What happens when we use a negative integer in this mini-program? Try it and see. Change your program to read as follows and RUN.

```
10 CLS:DEFINTA:A=-1000:L=VARPTR(A):PRINTL,L+1
20 PRINTPEEK(L),PEEK(L+1)
30 B=PEEK(L)+256*PEEK(L+1):PRINTB
```

```
OUTPUT:     17208          17209
            240            216
            55536
```

It is readily apparent that -10000 does not exactly = +55536. All integers are stored in RAM in "two's complement format" to allow us to use both positive and negative numbers within the allowable integer range of -32768 to +32767. So, how do we correct for this happenstance? Very easily. Undoubtedly you have already noted that +55536 is exactly +65536 MORE THAN our real integer of -10000, so all we need do is subtract 65536 IF our integer is a negative number.

There are all sorts of ways we may modify our mini-BASIC program to accomodate negative integers. The one we like best uses BASIC's SGN function because it reminds the programmer that we do indeed have a separate BASIC function just for determining the sign of a number; i.e., if SGN(X) = +1 it is positive, = zero it is zero, or = -1 it is negative. Add line 25 to your mini-BASIC program and modify line 30 as follows:

```
10 CLS:DEFINTA:A=-10000:L=VARPTR(A):PRINTL,L+1
20 PRINTPEEK(L),PEEK(L+1)
25 C=0:IFSGN(A)=-1THENC=65536
30 B=PEEK(L)+256*PEEK(L+1):PRINTB-C
```

```
OUTPUT:      17235          17236
              240            216
            -10000
```

By George, it works. If you wish to become more familiar with integers modify line 10 as shown below and try it with ANY negative or positive number you wish.

```
10 DEFINTA:INPUTA:L=VARPTR(A):PRINTL,L+1
```

Level II ROM will tell you in no uncertain terms when you have exceeded the allowable integer range. Try 32768, -32769, and 65535 for starters.

What happens when we divide 2 integers, forinstance 3 divided by 2? Level II ROM is smarter than the average bear. ALL integer divides OUTPUT in single precision floating point. What happens when we add, subtract, or multiply 2 integers whose result is greater than 32767 or less than -32768? Any overflow beyond integer range forces the output again, into single precision. Load the following brief program:

```
10 DEFINTA-C:INPUTA,B:D=A*B:IFD<-32768ORD>32767GOTO40
20 C=A*B:L=VARPTR(C):E=PEEK(L):F=PEEK(L+1)
30 PRINT"THE INTEGER =";E;"AND";F;"SPLIT DECIMAL.":GOTO10
40 L=VARPTR(D):E=PEEK(L):F=PEEK(L+1):G=PEEK(L+2):H=PEEK(L+3)
50 PRINT"SINGLE PRECISION FORMAT =";E;F;G;H:GOTO10
```

This program illustrates the "overflow" operation when the product of any 2 numbers YOU input is out of integer range, otherwise it outputs the familiar split decimal numbers for an integer. IF the product is out of integer range, VARPTR in line 40 gives the MEM location for a single precision number.

Page 8/9 in the Level II BASIC Reference Manual illustrates the format of single precision floating point output with the first number the LSB, the second number the next MSB, the third number the MSB, and the fourth number the exponent for normalized exponential notation. We will save deciphering any single precison numbers for Chapter 5. Try feeding this mini-program INPUTs of 2 times 16384, 4 times 16384, 8 times 16384, and 16 times 16384. Give you any ideas?. It certainly should.

LEVEL II VARIABLE IDENTIFICATION IN RAM:

The 3 bytes preceeding the MEM address returned by the VARPTR function give us the variable's identification using the following convention when L = VARPTR(variable):

```
L-3 = NT (number type); 2=integer 3=string 4=SNG 8=DBL
L-2 = 2nd character of variable name or zero if none.
L-1 = 1st character of variable name.
```

Try this mini-one liner program to see how it works:

```
10 INPUTAA:L=VARPTR(AA):PRINTPEEK(L-3),PEEK(L-2),PEEK(L-1)
```

It makes no nevermind what kind of number, integer, single, or double precision that we INPUT. The program will ALWAYS output

```
4 (=single precison)    65 (=ASCII A)    65 (=ASCII A)
```

UNLESS we tell it in our wisdom to do otherwise. Add line 5 to the program:

```
5 DEFINTA                          and RUN. Output now reads:

2 (=integer)    65 (=ASCII A)    65 (=ASCII A)
```

You should note that ANY variable beginning with 'A' is now defined as an integer. It could be 'AZ', 'A8', or 'AARDVARK' if you wished. If you defined the variable's name as AARDVARK, only the FIRST 2 letters be used. Try it.

Let's try the following NEW program on strings AND RUN.

```
10 INPUTAA$:L=VARPTR(AA$):PRINTPEEK(L-3),PEEK(L-2),PEEK(L-1)
```

You may input almost any gibberish you wish. OUTPUT should =

```
3 (=string)    65 (=ASCII A)    65 (=ASCII A)
```

So much for a variable's IDENT. Let's get back to integers.

Why not put all our integer knowhow together into a single program that will allow us to input the number, and then output from MEM its NT (number type), its variable name, and its value? Why not indeed. Practice makes perfect. Load the following program and RUN.

```
10 CLS:DEFINTA:INPUTAA:L=VARPTR(AA):A=PEEK(L-3)
20 IFA=2THENPRINT"AMAZING GRACE - IT IS AN INTEGER"
25 E=0:IFSGN(AA)=-1THENE=65536
30 B=PEEK(L-2):C=PEEK(L-1):PRINT"ITS VARIABLE NAME IS";B;C
40 D=PEEK(L)+256*PEEK(L+1):PRINT"ITS VALUE IS";D-E
```

This mini-program is quite obviously rigged; i.e., the duck always wins, but it nevertheless illustrates a number of the more important points covered in this Chapter regardless of whether or not YOU too bet on the duck.

As an encore to this superb performance, let's write our own VARPTR program that will find the variable AA most anywhere in memory for us and output its value on the video display. It is not the 'fastest gun in town' nor is it pluperfect, but it works quite well. Ok, it could conceivably be fooled by a DATA statement that included the series of: 2, 65, 65, 16, 39 which is a rare possibility that we will intentionally ignore.

```
10 DEFINTA:AA=10000
20 FORX=17129TO32767:B=PEEK(X)
30 IFB=2ANDPEEK(X+1)=65ANDPEEK(X+2)=65THENC=X+3ELSENEXT
40 PRINT"OUR VARPTR =";C:D=PEEK(C)+256*PEEK(C+1)
50 PRINT"AA'S VALUE =";D
>RUN
OUR VARPTR = 17276
AA'S VALUE = 10000
```

WISH TO DIG DEEPER ?
For those with real maschistic tendencies, take a look at the partially disassembled Level II ROM's CALL address for VARPTR beginning at 24EBH on page 69. Tracking it through its very own labyrinthine logic is even better than whips and cat o'nine tails. We suggest you unwind a ball of string along your path so you can find your way home. Seriously, VARPTR is an EASY ONE to unravel, comparatively speaking.

There are 3 ROM CALLs that are distant-distant cousins to the VARPTR function. For intrepid experimenters they are:

CALL 2337H:
String expression to ACCUM and NT to 40AFH. Requires string delimiter. Input address of string in HL register pair. It outputs string delimiter address in HL.

CALL 2540H:
Variable to ACCUM and NT to 40AFH. Input address of variable in HL register pair. It outputs HL register with value of NEXT character AFTER variable.

CALL 260DH:
Locate a variable, but if it DOES NOT exist create one and probably foul-up the location of ALL previously located variables. Use this CALL with CAUTION. Point HL to variable and hope. Outputs HL with value of next character after the variable name. This is the ROM CALL most similar to VARPTR.

The foregoing 3 CALLs are seldom used in assembly language programming and included ONLY for those who wish to write their own book/handbook about these CALLS' exquisitely dreary capabilities.

CONCLUSION:

All these pages on VARPTR integer location, conversion to decimal, and identification of the integer itself may seem rather pointless to the experienced assembly language buff who always skis the downhill 'straight down.' The reason we have chosen to beat the VARPTR camel to an inch of its life is that since Volume 1 was published we have received more questions that a thorough understanding of VARPTR would have answered than any other. As such, we're going to ride this tired old VARPTR camel for another two Chapters whether you hot-doggers like it or not.

IF you know everything about strings and string/arrays and KNOW HOW to convert single & double precision numbers stored in MEM to decimal with only a single line of BASIC, by all means be our guest and JP right now to Chapter 6. You may always use this copy of the Disassembled Handbook to swat flies or as a good 'starter' for green logs in the fireplace.

With these 'drop dead' comments, Chapter 3 now comes to a merciful end. You now have the opportunity to test yourself with the self-programmed learning questions for Chapter 3.

Another reminder: if you actually loaded and ran ALL the programs in this Chapter, that certain MAGIC that transpires when you do so will undoubtedly have given you a score that justifies your going on to Chapter 4.

Score yourself as follows for CORRECT answers:

```
100 percent = you understood EVERYTHING
 90 percent = you misunderstood something
 80 percent = you have poor retention and recall
 70 percent = you should re-study Level II BASIC
 60% or <   = re-read Doc Lien's Level I book
```

```
LD          A,(test score)
CP          100
JR          Z,(Chapter 4)
CP          90
JR          Z,(Chapter 4)
CP          80
JR          Z,(Chapter 3)
CP          70
JR          Z,(Level II)
CP          60
JR          Z,(Level I)
END         BEGIN
```

- CHAPTER 4 -

STRING & STRING ARRAY IDENTIFICATION, LOCATION, & RECALL

INTRODUCTION:

This is a short and easy Chapter that will clarify Level II's modus operandi for storing, identifying, and recalling strings & string arrays. It is considerably different from that used for storing integers and single/double precision floating point numbers. Again, for illustration, we will use Level II's VARPTR function and a few brief programs to show you exactly how these variables are stored in MEM. A picture is worth a thousand words, so we'll use as many pictures as is possible.

SOLITARY STRINGS:

You all know that Level II ROM automatically CLEARs 50 bytes for string storage every time we initialize (RUN) a BASIC program. If we wish more MEM space for for strings, we must CLEAR it. Conversely, if we do not wish any space for strings we may regain those 50 bytes by the statement, CLEAR 0.

As mentioned above, strings are stored quite DIFFERENTLY in MEM than integers or single/double precision numbers. The VARPTR function will help us illustrate this fact clearly. The only similarity between string storage in MEM and numbers is the way with which the string ID (identification) and string name are stored at L-3, L-2, & L-1 as shown 7 lines from the bottom of page 88; i.e., L=VARPTR(AA$):

L-3 = 3 (string NT) L-2 = 65 (ASCII A) L-3 = 65 (ASCII A)

The MEM address returned by L = the length of the string, and L+1 & L+2 return the address in MEM where the string is stashed away. Load and RUN the following short program to illustrate the last 3 points.

```
10 REM   - VARPTR STRING LOCATION PROGRAM -        VAR1
15 CLS
20 A$="THIS SURE BEATS WORKING FOR A LIVING.":L=VARPTR(A$)
25 PRINT"A$ LENGTH IS AT MEM LOCATION";L;
30 Y=PEEK(L):PRINT"AND IS";Y;"BYTES LONG.":PRINT
35 PRINT"A$ ADDRESS IN MEM IS AT LOCATIONS";L+1;"AND";L+2:PRINT
40 PRINT"A$ ADDRESS = ";
45 PRINT"PEEK(";L+1;") + 256 * PEEK(";L+2;")":PRINT
50 Z=(PEEK(L+1)+256*PEEK(L+2))
55 PRINT"A$ IS LOCATED IN MEM FROM";Z;"TO";Z+Y
60 PRINT:PRINT"A$ CONTAINS : ";
65 FORX=ZTOZ+Y-1:PRINTCHR$(PEEK(X));:NEXT
```

PROGRAM OUTPUT:

```
A$ LENGTH IS AT MEM LOCATION 27255 AND IS 37 BYTES LONG.

A$ ADDRESS IN MEM IS AT LOCATIONS 27256 AND 27257

A$ ADDRESS = PEEK( 27256 ) + 256 * PEEK( 27257 )

A$ IS LOCATED IN MEM FROM 26878 TO 26915

A$ CONTAINS : THIS SURE BEATS WORKING FOR A LIVING.
READY
>_
```

"JKL PRINTOUT"

If the MEM locations look strange to you non-disk programmers, it is because we are using NEWDOS+ and its convenient "JKL" feature to LPRINT this program's output from the video display. Please be patient. Together, we'll write a program in Chapter 6 that will give the non-disk user the same convenient option with a considerable improvement created by Bryan Mumford (of Mumford Micro) of 3 speed TRS-80 clock fame. Rather than LPRINTing all 16 lines of video (or having to turn the printer off), we will be able to EXIT the LPRINT video subroutine by simply hitting the space bar which will return us to whatever we were up to before pressing 'JKL' or '123', your choice. Now we'll run the SAME program in non-disk LEVEL II and take a look.

PROGRAM OUTPUT:

```
A$ LENGTH IS AT MEM LOCATIONS 17574 AND IS 37 BYTES LONG.

A$ ADDRESS IN MEM IS AT LOCATIONS 17575 AND 17576

A$ ADDRESS = PEEK ( 17575 ) + 256 * PEEK ( 17576 )

A$ IS LOCATED IN MEM FROM 17197 TO 17234

A$ CONTAINS : THIS SURE BEATS WORKING FOR A LIVING.
READY
```

"JKL PRINTOUT"

The table below reviews L = VARPTR(A$) with reference to L:

```
L-3 = NT (number type) = 3 for a string
L-2 = 2nd character of string or 0 if none
L-1 = first character of string variable
L   = byte length of string
L+1 = LSB of string address in MEM
L+2 = MSB of string address in MEM
```

STRING ARRAYS:

Are really not all that different from simple strings covered in the last 2 pages. The VARPTR function will return the string array's length location in MEM same as an ordinary string. Forinstance, L = VARPTR(A$(R)). L+1 and L+2 will give us the location in MEM where the string is stashed away exactly the same as an ordinary string.

Here is a considerably modified program from page 257 of the excellent text, "Learning Level II." Try loading it and RUNning it to get the feel for string arrays. String arrays are truly neat gizmos in that we can mix up alphabet, numbers and most any mish-mash except delimiters without worry or fret aforethought.

```
10 REM - VARPTR ADDRESS LOCATOR - AS MODIFIED BY W4UCH/2
20 CLS:FORR=1TO5:READA$(R)
30 PRINT"THE CHARACTER IS ";A$(R);". "
40 L=VARPTR(A$(R)):PRINT"ITS $ LENGTH =";PEEK(L);"IS AT";L
50 PRINT"IT IS REALLY AND TRULY LOCATED AT";(PEEK(L+1)+256*PEEK(L+2))
60 NEXT
70 DATA A,B,W60VP-TIP UR HAT TO HIM,1,2
80 GOTO80
```

PROGRAM OUTPUT:

```
THE CHARACTER IS A.
ITS $ LENGTH = 1 IS AT 17429
IT IS REALLY AND TRULY LOCATED AT 17362

THE CHARACTER IS B.
ITS $ LENGTH = 1 IS AT 17432
IT IS REALLY AND TRULY LOCATED AT 17364

THE CHARACTER IS W60VP-TIP UR HAT TO HIM.
ITS $ LENGTH = 23 IS AT 17435
IT IS REALLY AND TRULY LOCATED AT 17366

THE CHARACTER IS 1.
ITS $ LENGTH = 1 IS AT 17438
IT IS REALLY AND TRULY LOCATED AT 17390

THE CHARACTER IS 2.
ITS $ LENGTH = 1 IS AT 17441
IT IS REALLY AND TRULY LOCATED AT 17392
```

What does the above program's output tell us is the most significant difference between MEM storage of ordinary strings and string arrays? Well now, string array length and string array MEM locations' L+1 and L+2 are in their normal and ordinary locations, BUT since each string is spaced ONLY 3 bytes apart.....so where did our string idenitifier at L-3 and our string name(s) at L-2 and L-1 disappear to?

HINT: Take a look at those bytes in MEM immediately preceeding the first string array length location. "There's got to be a pony in there somewhere."

Can any of you readers figure out the secret masqued identity of W6OVP without an Amateur Radio Callbook? If you read the "User's Manual For Level I," and/or "Learning Level II," you should know him and his superb texts quite well. Of course, it is Dr. David A. Lien, Dean of Mathematics and Physical Sciences at Grossmount College near San Diego, California.

"Doff your hat to him in passing GO. Do collect $200."

Now take a breather and reflect upon how very easily string lengths and locations are snaked out of MEM using the almost ALL PURPOSE VARPTR function.

Ok, the break is over. You should be able to answer the self-programmed learning questions for this Chapter virtually blindfolded they are so easy. It is impossible to write difficult questions about such a straightforward and simple subect.

The next Chapter has everything you ever wanted to know about deciphering single & double precision floating point numbers. Though the subject sounds gruesome at best, it is so simple that you will not believe it until you try it.

A single line of simple Level II BASIC will decipher ANY single precision, double precision, or even quadruple precision floating point number. Turn the page and have a GO at doing it. "You WILL enjoy it," the Lufthansa pilot said.

NOTES:

- CHAPTER 5 -

SINGLE & DOUBLE PRECISION FLOATING POINT NUMBERS

INTRODUCTION:

For many years there has been an aura and mystique surrounding single and double precision, floating point notated, normalized exponential numbers that is totally undeserved. With a name like that it is not surprising. It may be the name that terrifies? The problem appears to have 3 origins:

1. Many gifted writers do not fully understand the subject.

2. Those gifted individuals who do understand the subject are not competent writers.

3. Those few competent writers that understand the subject choose to cloak their treatises explaining the subject in jargon understood only by fellow academics in academia.

When we finish this Chapter you will understand how to decode most any variety of single and double precision floating point number and use them as you wish. Again, we will use a few very simple basic BASIC programs in our TRS-80 to unravel this heretofore seemingly mysterious black hole. By the time we are finished, it will be readily apparent to you that, "the King truly has no clothes on at all."

SINGLE & DOUBLE PRECISION NUMBER STORAGE IN MEM:

Is one of the major reasons why Microsoft's Paul Allen and Bill Gates selected these formats for Level II. It sure is a conveniently compact way of storing BIG numbers in SMALL spaces. Remember that MEM costs money and money = cost effectiveness of whatever product you are peddling. Another major reason for this format is the ease with which our Z-80 may be programmed for single and double precision + - * /. After all, the only arithmetic function the fundamental micro-processor can accomplish is ADDING 2 numbers together. The other arithmetic functions of subtraction, multiplication, and division are naught but variations on the theme of the addition process.

Its time to get our feet wet. We shall enter the water and the subject ever so carefully. We have never lost a swimmer here at the 'ole swimming hole. Load and RUN the following program:

```
10 DEFSNGA:INPUTAA:K=VARPTR(AA)
20 PRINT"K-3 =";PEEK(K-3):PRINT"K-2 =";PEEK(K-2):PRINT"K-1 =";PEEK(K-1)
30 PRINT"K   =";PEEK(K):PRINT"K+1 =";PEEK(K+1):PRINT"K+2 =";PEEK(K+2)
40 PRINT" K+3 =";PEEK(K+3)
50 GOTO10
```

Let's start off by using the following round powers of 2:

	INPUT 1	INPUT 2	INPUT 4	INPUT 8	INPUT 16
K-3 =	4	4	4	4	4
K-2 =	65	65	65	65	65
K-1 =	65	65	65	65	65
K =	0	0	0	0	0
K+1 =	0	0	0	0	0
K+2 =	0	0	0	0	0
K+3 =	129	130	131	132	133

We readily observe that VARPTR again (as in Chapter 3) returns 4 = single precison at K-3, 65 = ASCII 'A' at K-2, and 65 = ASCII 'A' at K-1 to annotate and identify our single precision variable AA. The following 4 MEM locations contain our now compacted number, hopefully accurate out to 6 significant digits with the last digit rounded off.

K = the least significant byte (LSB)
K+1 = the next most significant byte (NMSB)
K+2 = the most significant byte (MSB)
K+3 = the exponent to the power of 2 (see below)

It does not require a high mathematical aptitude to note that if we subtract 129 from K+3 we obtain the exact power of 2 that is = to the number we INPUT; i.e., 2 ↑ 0 = 1, 2 ↑ 1 = 2, 2 ↑ 2 = 4, 2 ↑ 3 = 8 and 2 ↑ 4 = 16.

So far, so good. Let's DELETE line 20 since we know how to locate and identify our variable and change line 50 to read as below & RUN:

```
50 N=PEEK(K+3)-129:PRINT"NO. =";2↑N:GOTO10
```

	INPUT 1	INPUT 2	INPUT 4	INPUT 8	INPUT 16
K =	0	0	0	0	0
K+1 =	0	0	0	0	0
K+2 =	0	0	0	0	0
K+3 =	129	130	131	132	133
NO. =	1	2	4	8	16

Have we solved our single precision decoding problem already? Answer: sure we have IF it is a nice round power of 2 and it is a POSITIVE number. Try the following numbers which are all nice round powers of 2: 32, 64, 128, 256, 512, and so forth.

Ok, how about numbers LESS THAN 1 and greater than zero? It makes no nevermind IF the numbers we INPUT are nice round NEGATIVE powers of 2. After all 2 ↑ -1 = .5, 2 ↑ -2 = .25, 2 ↑ -3 = .125, 2 ↑ -4 = .0625 and 2 ↑ -5 = .03125. Try it and see what happens.

	INPUT.5	INPUT.25	INPUT.125	INPUT.0625	INPUT.03125
K	= 0	0	0	0	0
K+1	= 0	0	0	0	0
K+2	= 0	0	0	0	0
K+3	= 128	127	126	125	124
NO.	= .5	.25	.125	.0625	.03125

Amazing, it works. For some really small numbers try these: 7.8125E-03 and 3.90625E-03. Both are negative powers of 2.

For those enthralled with both negative and positive powers of 2, load and RUN the following program. The reason that line 20 has A=A*2 rather than the 'up-arrow' exponent sign is that many of the early TRS-80 Level II ROM chips were not perfectly shadow-masked at the time of manufacture and as such the exponent sign is prone to occasional errors when used. A=A*2 eliminates these errors.

```
10 DEFSNGA:INPUTA:PRINT" NUMBER","    STORED IN MEM"
20 PRINT" X 2  ","    K  K1 K2 K3"
30 A=A*2:PRINTA,"=  ";:L=VARPTR(A):C=L+3
40 FORX=LTOC:PRINTPEEK(X);:NEXT:PRINT:GOTO30
50 REM TRY INPUT  .015625
```

PROGRAM OUTPUT:

NUMBER X 2	STORED IN MEM K	K1	K2	K3
.03125	0	0	0	124
.0625	0	0	0	125
.125	0	0	0	126
.25	0	0	0	127
.5	0	0	0	128
1	0	0	0	129
2	0	0	0	130
4	0	0	0	131
8	0	0	0	132
16	0	0	0	133
32	0	0	0	134
64	0	0	0	135
128	0	0	0	136
256	0	0	0	137
512	0	0	0	138
1024	0	0	0	139
2048	0	0	0	140
4096	0	0	0	141
8192	0	0	0	142
16384	0	0	0	143
32768	0	0	0	144
65536	0	0	0	145
131072	0	0	0	146
262144	0	0	0	147
524288	0	0	0	148
1.04858E+06	0	0	0	149
2.12676E+37	0	0	0	253
4.25353E+37	0	0	0	254

OVERFLOW IN 30

The - - - - - - dotted line is where we interrupted this mini-program to save a few pages of LPRINT out. The OVERFLOW at the end signifies we have a number TOO large for the TRS-80 to handle and in most cases, interplanetary travel excepted, so large as to be meaningless.

EXPONENT SIGN ACCURACY TEST FOR LEVEL II ROM:

Since we mentioned our Level II ROM's exponent sign accuracy problem on the last page, let's take a brief side trip to investigate whether your TRS-80 has the same problem. Try these two 1 line programs on your TRS-80:

```
10 CLS:DEFDBLA:A=1:FORX=1TO44:A=A*2:PRINTA,:NEXT
```

PROGRAM OUTPUT:

```
 2                 4                 8                 16
 32                64                128               256
 512               1024              2048              4096
 8192              16384             32768             65536
 131072            262144            524288            1048576
 2097152           4194304           8388608           16777216
 33554432          67108864          134217728         268435456
 536870912         1073741824        2147483648        4294967296
 8589934592        17179869184       34359738368       68719476736
 137438953472      274877906944      549755813888      1099511627776
 2199023255552     4398046511104     8796093022208     17592186044416
```

```
10 CLS:FORX=1TO44:A#=VAL(STR$(2↑X)):PRINTA#,:NEXT
```

PROGRAM OUTPUT:

```
 2                 4                 8                 16
 32                64                128               256
 512               1024              2048              4096
 8192.01           16384             32768             65536
 131072            262144            524288            1048580
 2097150           4194300           8388610           16777200
 33554400          67109000          134218000         268435008
 536870976         1073740032        2147480064        4294969856
 8589930496        17179901952       34359701504       68719403008
 137439019008      274878038016      549756076032      1099510120448
 2199020240896     4398049656832     8796090400768     17592198627328
```

If BOTH of your print outs are identical to the FIRST program output, you do not have a problem as this is perfect. If your second program output is NOT identical to the first one, back to the Radio Shack doctor, or learn to live with it as we have. It is not all that difficult to work around, as long you know it is there (try Listerine).

SINGLE PRECISION NEGATIVE NUMBER STORAGE CONVENTION:

The first bit of our number's MSB (most significant byte) is the positive or negative signpost. Since all our programs so

far in this Chapter have dealt with nice round powers of 2, (both - and + powers), MEM location K+2 (K2) has always been decimal zero. If any of the numbers used so far had been NEGATIVE, K+2 would have been decimal 128 = binary 10000000. The leading '1' of the MSB's (most significant byte) most significant bit is that signpost. IF we mask that most significant bit of K+2 by subtracting 128 from the value of K+2 IF and when K+2 is = to or greater than 128, THEN ALL NEGATIVE SINGLE PRECISION NUMBERS ARE STORED EXACTLY THE SAME WAY AS POSITIVE NUMBERS.

Let's try the following program again, but with negative nos.:

```
10 DEFSNGA:INPUTAA:K=VARPTR(AA)
20 PRINT"K   =";PEEK(K):PRINT"K+1 =";PEEK(K+1)
30 PRINT"K+2 =";PEEK(K+2):PRINT"K+3 =";PEEK(K+3)
40 GOTO10
```

	INPUT-1	INPUT-2	INPUT-4	INPUT-8	INPUT-16
K	= 0	0	0	0	0
K+1	= 0	0	0	0	0
K+2	= 128	128	128	128	128
K+3	= 129	130	131	132	133

If we look back at the bottom of page 96, we note that K+3 is EXACTLY the same value for the NEGATIVE numbers as it was for the same POSITIVE numbers. Only K+2's first bit is a 1 = 10000000 = 128 decimal signifying that the number is indeed NEGATIVE. Now let us add some smarts to the above program that will test the value of K+2 and IF it => 128 then prints out its value as a negative number.

```
10 DEFSNGA:INPUTAA:K=VARPTR(AA)
20 PRINT"K   =";PEEK(K):PRINT"K+1 =";PEEK(K+1)
30 PRINT"K+2 =";PEEK(K+2):PRINT"K+3 =";PEEK(K+3)
40 S=1:IFPEEK(K+2)=>128THENS=-1
50 N=PEEK(K+3)-129:PRINT"NO. =";2↑N*S:GOTO10
```

PROGRAM OUTPUT:

	INPUT-1	INPUT-2	INPUT-4	INPUT-8	INPUT-16
K	= 0	0	0	0	0
K+1	= 0	0	0	0	0
K+2	= 128	128	128	128	128
K+3	= 129	130	131	132	133
NO.	=-1	-2	-4	-8	-16

Be patient. We're almost there. Just for the sake of you purists a VERY short discussion on normalization and then we'll GOTO the big bazoo on how to decode single & double precision, floating point, normalized exponential numbers.

Like all supposedly complicated difficult gizmos, it is really nothing more than a group of absurdly simple principles which only LOOK complex when taken as a whole.

FLOATING POINT NORMALIZATION:

Is much like motherhood and the flag. Good clear through. Its most important advantages are:

1. It substantially increases the accuracy of our TRS-80's arithmetic calculations by STASHING away more significant digits per unit memory than any other storage routine.

2. This storage format works extremely efficiently with ALL the other single/double precision arithmetic subroutines.

First things first. You will recall that the first bit of our most significant BYTE is our positive/negative signpost. IF it is zero it = a positive number, and IF a 1 it = a negative number. As such we have 23 bits left to play with; i.e., 7 in the most significant byte, plus 8 in the next most significant byte and another 8 in the least significant byte = 23 total.

The first thing our Level II ROM checks is to see if they are all zeros, as they ALL were in the last few pages. If so, it skips the normalization procedure entirely as it is completely unnecessary. If any of these 23 bits are NOT zero, OFF IT GOES to the floating point normalization routine as follows:

1. IF THE NUMBER WAS ORIGINALLY= 0.00001111000011110000111 E+0

2. AFTER NORMALIZATION IT WILL = 0.11110000111100001110000 E-4

All the normalization accomplished was to move the binary number to the LEFT four places so that the FIRST 1 in the number was moved up to the implied decimal point. For each bit position that is shifted left, the exponent is decremented by 1 so that correct magnitude of the number is maintained. It is MOST IMPORTANT to note that the number's normalized value will now fall BETWEEN 1/2 and 1. More later on this important 1/2 when we convert it to decimal.

The Level II ROM normalization process is organized ONLY to work with POSITIVE numbers. So what happens when the number is NEGATIVE? Why, back to the old two's complement routine which we will not bore you with. THEN the number is normalized, and then complemented AGAIN. Do not knock it if it works, and it works very well indeed.

WISH TO DIG DEEPER?

The best write up on this subject we have seen only deserves our good rating. The rest of the write ups we have seen vary from HORRID to IMPOSSIBLE. If you truly wish to immerse yourself in the subject we suggest you buy a wet suit, and then a SCUBA tank or life jacket and try:

"Z-80 Software Gourmet Guide & Cookbook," by Nat Wadsworth. published by Scelbi Publications...dated August 1979.

DECODING SINGLE & DOUBLE PRECISION NUMBERS:

We will make a frontal attack on single precision first, in this section, and then leap-frog the niceties straight into double precision as the decipherment is just as simple as was single precision. If Level II BASIC can encode it, it is rather obvious that together we can write a BASIC program to decode it. The ONLY DIFFERENCE is that hopefully our decoding program is straightforward and understandable to the average bear, whereas previous explanations somehow or other got side-tracked off to the the far side of the moon.

Take a look at the following simple equation which is going to very efficiently do the decoding job for us:

$$N = 2\uparrow(K3-128) * .5 + \frac{K2+\frac{K1+\frac{K}{256}}{256}}{256}$$

The above equation will WORK for any positive single precision number that Level II can handle. For negative numbers we need only test K2 to see if it => 128 and multiply N by a + or - number as appropriate (which we already did on page 99). There is the 1/2 = .5 we mentioned earlier. The number to the right of the * sign will ALWAYS be => .5 and less than 1.

Now let's reduce all these good things we have covered in this Chapter to a real working, do something worthwhile, program.

```
10 ' SINGLE PRECISION/FLOATING POINT DECODING DEMO BY W4UCH/2
15 DEFINTC-M,O-Z:DEFSNGA:B=1:S=1
20 INPUT"INPUT ANY SINGLE PRECISION NUMBER";AA:L=VARPTR(AA)
25 K=PEEK(L):K1=PEEK(L+1):K2=PEEK(L+2):K3=PEEK(L+3)
30 PRINT" K"," K1"," K2"," K3":PRINTK,K1,K2,K3:E=K3-128
35 IFK2=>128THENK2=K2-128:S=-1
40 B=2↑E
50 N=(B)*(.5+((K2+((K1+(K/256))/256))/256)):PRINTN*S:GOTO15
```

We will run through the program briefly. Line 15's DEFINT is unnecessary, but speeds up execution time. Variable B is initialized at 1 and variable S, our SIGN variable at plus 1, so we may repeat the program with the GOTO in the last line.

Lines 20 - 40, we are familiar with from the preceeding pages.

Line 50 is the above simple single precision/floating point decoding equation in algebraic format. The extra parentheses are included for clarity. As in this Chapter's previous programs, the variable S simply tells the program whether the output is positive or negative. Bet you never thought it would be this simple.

If the exponent test program on page 98 showed that you too have an accuracy problem when using this function, by all means use the modified program on the next page.

EXPONENT FUNCTION PROBLEM "END RUN" SOLUTION:

```
10 ' SINGLE PRECISION/FLOATING POINT DECODING DEMO BY W4UCH/2
15 DEFINTC-M,O-Z:DEFSNGA:B=1:S=1
20 INPUT"INPUT ANY SINGLE PRECISION NUMBER";AA:L=VARPTR(AA)
25 K=PEEK(L):K1=PEEK(L+1):K2=PEEK(L+2):K3=PEEK(L+3)
30 PRINT" K"," K1"," K2"," K3":PRINTK,K1,K2,K3:E=K3-128
35 IFK2=>128THENK2=K2-128:S=-1
40 IFE=0GOTO50ELSEIFE=>1THENFORZ=1TOE:B=B*2:NEXT:GOTO50
45 E=E*-1:FORZ=1TOE:B=B/2:NEXT
50 N=(B)*(.5+((K2+((K1+(K/256))/256))/256)):PRINTN*S:GOTO15
```

Following are a few RUNs:

```
INPUT ANY SINGLE PRECISION NUMBER? .999999
 K                K1               K2               K3
 240              255              127              128
 .999999
INPUT ANY SINGLE PRECISION NUMBER? 10000000000000000
 K                K1               K2               K3
 202              27               14               182
 1E+16
INPUT ANY SINGLE PRECISION NUMBER? .0000000000000001
 K                K1               K2               K3
 147              149              102              75
INPUT ANY SINGLE PRECISION NUMBER? 1111111
 K                K1               K2               K3
 56               162              7                149
 1.11111E+06
INPUT ANY SINGLE PRECISION NUMBER? .1111111
 K                K1               K2               K3
 56               142              99               125
 .111111
INPUT ANY SINGLE PRECISION NUMBER? 123456
 K                K1               K2               K3
 0                32               113              145
123456
INPUT ANY SINGLE PRECISION NUMBER? -.123456
 K                K1               K2               K3
 126              214              252              125
-.123456
INPUT ANY SINGLE PRECISION NUMBER? -90000000000000000
 K                K1               K2               K3
 158              203              255              181
-9E+15
INPUT ANY SINGLE PRECISION NUMBER? -.000000000000001
 K                K1               K2               K3
 124              29               144              79
-1E-15
INPUT ANY SINGLE PRECISION NUMBER? -.999999
 K                K1               K2               K3
 240              255              255              128
-.999999
```

The "end run" solution to the exponent problem on page 102 is straightforward enough. For those interested, note lines 40 and 45 which perform the same function for BOTH positive and negative sign exponents. IF the exponent is zero, 2↑0 = 1, then the first statement in line 40 GOTOs line 50 as variable 'B' was already initialized at '1' in line 15. The rest of line 40 puts variable B=B*2 in the middle of a FOR-NEXT loop for the number of times = to the power that we wish to raise 2 to. Line 45 is similar, but for negative powers of 2. The only difference is that now the program divides, B=B/2, for the number of times which we wish to raise 2 to the negative power of 2. It is not terribly sophisticated, but it works very well, thank you.

There is something fascinating about these simple multiple precision-normalized-floating-point routines. If you wish to have some fun and have nothing better to do, try writing an encoding and decoding routine for 32 significant digits. It actually is quite easy. When you are finished, your little TRS-80 will THINK it has a Z8000 or MC68000 inside it. Maybe not quite so FAST, but with certainly the same accuracy. It may surprise some professionals in the field, but a TRS-80 can be made to emulate most any of the new programmable array processors IF speed is not the primary concern. Pipelining? Two TRS-80s can do that too if you have a little imagination.

DECODING DOUBLE PRECISION-FLOATING POINT NUMBERS - AT LAST:

Double precision storage in MEM is very little different from that of single precision EXCEPT that now we will use 8 bytes instead of 4. Double precision's K7 is the exponent byte the same as single precision's K3. K6's most significant BIT is the sign bit in double precision, same as K2's first BIT in single precision. Here is the decoding formula...familiar?

$$N=2\uparrow(K7-128) * .5 + \cfrac{K}{256 + \cfrac{K+1}{256 + \cfrac{K+2}{256 + \cfrac{K+3}{256 + \cfrac{K+4}{256 + \cfrac{K+5}{256 + \cfrac{K+6}{256}}}}}}}$$

Just like our single precision decoding equation, the above equation will WORK for any positive number that Level II is capable of handling. K again, is the least significant byte, K+1 the next most significant byte, K+2 the next most significant byte and so on, up to K+6, which is the most significant byte, plus having its first bit represent the sign of the number, 1 = negative and 0 = positive. Same as in the single precision decoding program, we'll test K+6 to see if => than 128 = 10000000 binary, and if so, adjust N's sign accordingly.

Now let's reduce this equation to a real-life hard-working double precision decoding program that really WORKS.

DOUBLE PRECISION DECODING USING THE EXPONENT FUNCTION:

```
10 ' DOUBLE PRECISION/FLOATING POINT DECODING DEMO BY W4UCH/2
15 DEFDBLA-Y:B=1:S=1:INPUT"INPUT ANY NUMBER CONCEIVABLE";AA
20 L=VARPTR(AA)
25 K=PEEK(L):K1=PEEK(L+1):K2=PEEK(L+2):K3=PEEK(L+3)
30 K4=PEEK(L+4):K5=PEEK(L+5):K6=PEEK(L+6):K7=PEEK(L+7)
35 PRINT"K  =";K:PRINT"K1 =";K1:PRINT"K2 =";K2:PRINT"K3 =";K3
40 PRINT"K4 =";K4:PRINT"K5 =";K5:PRINT"K6 =";K6:PRINT"K7 =";K7
45 IFK6=>128THENK6=K6-128:S=-1
50 E=INT(K7-128)
55 B=2↑E
65 N=(B)*(.5+((K6+((K5+((K4+((K3+((K2+((K1+(K/256))/256))/256))/
256))/256))/256))/256))
70 PRINTN*S:GOTO15
```

DOUBLE PRECISION DECODING WITHOUT USING THE EXPONENT FUNCTION:

```
10 ' DOUBLE PRECISION/FLOATING POINT DECODING DEMO BY W4UCH/2
15 DEFDBLA-Y:B=1:S=1:INPUT"INPUT ANY NUMBER CONCEIVABLE";AA
20 L=VARPTR(AA)
25 K=PEEK(L):K1=PEEK(L+1):K2=PEEK(L+2):K3=PEEK(L+3)
30 K4=PEEK(L+4):K5=PEEK(L+5):K6=PEEK(L+6):K7=PEEK(L+7)
35 PRINT"K  =";K:PRINT"K1 =";K1:PRINT"K2 =";K2:PRINT"K3 =";K3
40 PRINT"K4 =";K4:PRINT"K5 =";K5:PRINT"K6 =";K6:PRINT"K7 =";K7
45 IFK6=>128THENK6=K6-128:S=-1
50 E=INT(K7-128)
55 IFE=0GOTO65ELSEIFE=>1THENFORZ=1TOE:B=B*2:NEXT:GOTO65
60 E=E*-1:FORZ=1TOE:B=B/2:NEXT
65 N=(B)*(.5+((K6+((K5+((K4+((K3+((K2+((K1+(K/256))/256))/256))/
256))/256))/256))/256))
70 PRINTN*S:GOTO15
```

Except for the additional 4 bytes of MEM storage, double precision decoding is virtually identical to single precision. VARPTR still points to the least significant byte 'K' and our primary decoding line, line 65, is sure familiar except for the additional 4 MEM locations.

If you followed the logic and flow through the single precision decoding program, the only confounding conundrum that double precision decoding presents you with is, "is how absurdly simple BOTH procedures are when BASIC is used to illustrate the major points."

Let's take a look at a couple of pages of this program's output. We'll then FINALLY finish this long, but rather fascinating Chapter with a few comments about the similarity of number storage in MEM and the ACCUM and 'CS', the CDBL Store, plus review the ACCUM and CS storage locations for integers, single precision, and double precision numbers.

PROGRAM OUTPUT: "DECODING DOUBLE PRECISION NUMBERS"

```
INPUT ANY NUMBER CONCEIVABLE? 1.0
K  = 0
K1 = 0
K2 = 0
K3 = 0
K4 = 0
K5 = 0
K6 = 0
K7 = 129
 1

INPUT ANY NUMBER CONCEIVABLE? -1
K  = 0
K1 = 0
K2 = 0
K3 = 0
K4 = 0
K5 = 0
K6 = 128
K7 = 129
-1

INPUT ANY NUMBER CONCEIVABLE? 100000000000000000000000000000000
K  = 183
K1 = 112
K2 = 43
K3 = 168
K4 = 173
K5 = 197
K6 = 29
K7 = 235
 1D+32

INPUT ANY NUMBER CONCEIVABLE? .9999999999999999
K  = 250
K1 = 255
K2 = 255
K3 = 255
K4 = 255
K5 = 255
K6 = 127
K7 = 128
 .9999999999999999

INPUT ANY NUMBER CONCEIVABLE? .00000000000000000000000000000001
K  = 149
K1 = 57
K2 = 69
K3 = 173
K4 = 30
K5 = 177
K6 = 79
K7 = 22
 1D-32

INPUT ANY NUMBER CONCEIVABLE? 1234567890987654
K  = 192
K1 = 80
K2 = 253
K3 = 146
K4 = 167
K5 = 90
K6 = 12
K7 = 179
 1234567890987654
```

PROGRAM OUTPUT: "DECODING DOUBLE PRECISION NUMBERS"

```
INPUT ANY NUMBER CONCEIVABLE? 1.329227995784916E+36
K  = 0
K1 = 0
K2 = 0
K3 = 0
K4 = 0
K5 = 0
K6 = 0
K7 = 249
 1.329227995784916D+36

INPUT ANY NUMBER CONCEIVABLE? 6.103519990574569E-5
K  = 0
K1 = 0
K2 = 0
K3 = 0
K4 = 6
K5 = 0
K6 = 0
K7 = 115
 6.103519990574569D-05

INPUT ANY NUMBER CONCEIVABLE? 4096
K  = 0
K1 = 0
K2 = 0
K3 = 0
K4 = 0
K5 = 0
K6 = 0
K7 = 141
 4096

INPUT ANY NUMBER CONCEIVABLE? -4096
K  = 0
K1 = 0
K2 = 0
K3 = 0
K4 = 0
K5 = 0
K6 = 128
K7 = 141
-4096

INPUT ANY NUMBER CONCEIVABLE? .1111111111111111
K  = 221
K1 = 56
K2 = 142
K3 = 227
K4 = 56
K5 = 142
K6 = 99
K7 = 125
 .1111111111111111
```

ACCUMULATOR & CDBL STORE MEM LOCATIONS AND FORMAT:

You will recall from Volume 1's Chapter 2, that the ACCUM and CS (CDBL Store) in MEM were used by all the double precision assembly language arithmetic operations. They were:

ADDITION	:	ACCUM + CS
SUBTRACTION	:	ACCUM - CS
MULTIPLICATION	:	ACCUM * CS
DIVISION	:	ACCUM / CS

For single precision arithmetic functions, the ACCUM and BCDE registers were the workhorses, with the results always stored in the ACCUM for BOTH single and double precision arithmetic.

ADDITION	:	BCDE + ACCUM
SUBTRACTION	:	BCDE - ACCUM
MULITPLICATION	:	BCDE * ACCUM
DIVISION	:	BCDE / ACCUM

Integer arithmetic uses the ACCUM for three purposes:

1. All integer divides are output in single precision.
2. All adds, subtracts, and multiplies results that fall outside of the -32768 and +32767 integer range are output in single precision.
3. All integer arithmetic outputs are stored in the ACCUM.

It should come as no surprise that the number format for both ACCUM and CS number storage are EXACTLY the same as in Level II MEM except for the variable's NT and identification. Every exception has an exception. This one is: if you PEEK three MEM locations BEFORE the ACCUM you will find the NT stored there IF the DEFINT, DEFSNG, or DEFDBL A-Z statement is used.

The following are tables of ACCUM and CS MEM locations for single and double precision numbers in normalized floating point format:

- ACCUMULATOR -

DECIMAL ADDRESS	HEX ADDRESS	SINGLE PRECISION	DOUBLE PRECISION
16669	411D		LSB
16670	411E		NMSB
16671	411F		NMSB
16672	4120		NMSB
16673	4121	LSB	NMSB
16674	4122	NMSB	NMSB
16675	4123	MSB + SIGN	MSB + SIGN
16676	4124	EXPONENT	EXPONENT

	- CDBL STORE -		
DECIMAL ADDRESS	HEX ADDRESS	SINGLE PRECISION	DOUBLE PRECISION
16679	4127	LSB	LSB
16680	4128	NMSB	NMSB
16681	4129	MSB + SIGN	NMSB
16682	412A	EXPONENT	NMSB
16683	412B		NMSB
16684	412C		NMSB
16685	412D		MSB + SIGN
16686	412F		EXPONENT

ACCUM DEMONSTRATION EXERCISE:

After the last dozen pages, this program is downtown Simpleville. Here is a mini-program written in BASIC that is much like the programs at the beginning of this Chapter, EXCEPT now we will be looking at only the MSB and EXP of the ACCUM instead of our number stored in MEM. As the ACCUM table on the last page illustrates, the ACCUM's most significant byte is located at 16675 and the EXP at 16676 for both single and double precision numbers.

Writing an ACCUM demonstration program in BASIC is an interesting challenge as BASIC per se, uses the ACCUM for every arithmetic and string function imaginable. It is like trying to hit a very fast moving target in that by the vary act of assigning a variable a value, the contents of the ACCUM are usually changed in the process. As such, we will use a bit of poetic license in the ACCUM demo program; i.e., we will CHEAT.

The rules of this game are somewhat similar to a few of the early programs in this Chapter: we are only allowed to INPUT nice round numbers that are powers of 2. Numbers that are POSITVE or NEGATIVE powers of 2 ARE nevertheless allowed in this game. Forinstance, 2 ↑ 5 = 32; + or - 32 is an ok number and 2 ↑ -5 = .03125; + or - .03125 is also an ok number. The reason we are NOT using the ACCUM's least significant byte and next most significant byte is that these two MEM locations return the decimal equivalent of 16673 when we PEEK these two locations. The first location is indeed 16673. A mirror image of a mirror image of a mirror image, etc.? Probably not. This is the ACCUM's pointer that tells BASIC, "HERE I AM, USE ME."

```
10 'ACCUM DEMONSTRATION EXERCISE - ACM:1
15 CLS:A=16384:INPUT"INPUT = 16384 * ";B:C=A*B:PRINT
20 PRINT"ACCUM =","  MSB","  EXP"
25 PRINT,PEEK(16675),PEEK(16676):PRINT
30 S=1:IFSGN(C)=-1THENS=-1:PRINT"THE NUMBER IS NEGATIVE":C=C*-1
35 IFC=0THENPRINT"THE PRODUCT IS ZERO - TRY AGAIN":GOTO60
40 IFS=1THENPRINT"THE NUMBER IS POSITIVE AS THE MSB = < 128"
45 PRINT:X=LOG(C)/LOG(2)+129
50 PRINT"THE ACCUM EXP =";X;" AND";X;"-129 =";X-129:PRINT
55 X=X-129:PRINT"THE PRODUCT IS 2 ↑";X;" = ";2↑X*S
60 INPUTQ:GOTO15
```

PROGRAM LOGIC AND FLOW:

We'll attack this rascal line by line. As mentioned earlier, we had to CHEAT a lot to make it work. About the only truly HONEST segments of the program are the PEEK values printed out on your video display.....NO, we did not secret and hide any POKE values anywhere in the program to make it work, though it would have probably been easier to do so.

```
10 'A TITLE SO WE WOULD NOT LOSE OR MISPLACE THE PROGRAM.

15 'SINCE WE DID NOT DEFINE AN INTEGER OR DOUBLE PRECISION
    NUMBER, THE TRS-80 WILL AUTOMATICALLY ASSUME SINGLE
    PRECISION. THIS IS A SIMPLE MULTIPLICATION PROGRAM.  ANY
    NUMBER YOU INPUT * 16384.  C=A*B FORCES THE ACCUM TO DO
    SOME WORK.

20 'COLUMN HEADINGS

25 'PEEK VALUES OF ACCUM'S MSB AND EXP LOCATIONS IN MEMORY
    AND PRINT THEM OUT ON YOUR VIDEO DISPLAY.

30 'FROM HERE ON, THE PROGRAM IS ALL WINDOW DRESSING.  WE
    WILL RECONSTRUCT THE PRODUCT OF THE NUMBER YOU INPUT
    * 16384 USING SIMPLE BASIC FUNCTIONS MUCH LIKE THE WAY
    BASIC DISASSEMBLES THE NUMBER AND STORES IT IN THE ACCUM.
    THIS LINE'S ONLY FUNCTION IS TO DETERMINE THE SIGN & SET
    C & S APPROPRIATELY AND IF NEGATIVE, TELL YOU SO.

35 'A ZERO TRAP TO AVOID /ZERO ERROR.  THIS IS MUCH LIKE THE
    NORMALIZATION ROUTINE IN ROM THAT 1ST TESTS FOR ZERO.

40 'TELL YOU IF THE PRODUCT IS POSITIVE.

45 'LOG TO THE BASE 2 OF 'C' + 129.  SAME AS EXP PEEK VALUE.

50 'PRINT OUT THIS VALUE AND SUBTRACT 129.

55 'RAISE 2 TO THE POWER OF SUBTRACTED VALUE AND PRINT IT OUT
    ON YOUR VIDEO DISPLAY WITH THE CORRECT SIGN BY * S.

60 'A PAUSE TO ALLOW YOU TO ASSIMILATE THIS SPLENDID AND
    SURPRISING VIRTUOSO PERFORMANCE 'YOU' ACCOMPLISHED.
```

CONCLUSION: (meaning the end of this Chapter)

We've covered a lot of territory in this Chapter. Have a 'GO' at the questions for this Chapter to test your understanding and retention. If you DID NOT ENTER and RUN each and every little mini-program in the Chapter, the only one you have cheated is yourself.

The following page has a few examples of the ACCUM DEMONSTRATION PROGRAM'S output with both normal and weird numbers.

- ACCUM DEMONSTRATION EXERCISE -

PROGRAM OUTPUT:

```
INPUT = 16384 * ? 6.10352E-05

ACCUM =          MSB              EXP
                 0                129

THE NUMBER IS POSITIVE AS THE MSB = < 128

THE ACCUM EXP = 129  AND 129 -129 = 0

THE PRODUCT IS 2 ↑ 0  =  1


INPUT = 16384 * ? -6.10352E-05

ACCUM =          MSB              EXP
                 128              129

THE NUMBER IS NEGATIVE

THE ACCUM EXP = 129  AND 129 -129 = 0

THE PRODUCT IS 2 ↑ 0  = -1


INPUT = 16384 * ? 64

ACCUM =          MSB              EXP
                 0                149

THE NUMBER IS POSITIVE AS THE MSB = < 128

THE ACCUM EXP = 149  AND 149 -129 = 20

THE PRODUCT IS 2 ↑ 20  =  1.04858E+06


INPUT = 16384 * ? -64

ACCUM =          MSB              EXP
                 128              149

THE NUMBER IS NEGATIVE

THE ACCUM EXP = 149  AND 149 -129 = 20

THE PRODUCT IS 2 ↑ 20  = -1.04858E+06
```

- CHAPTER 6 -

SIMULATING 'JKL' LPRINT VIDEO FOR NON-DISK SYSTEMS

INTRODUCTION:

NEWDOS+ and NEWDOS 80 are chock full of goodies for the long suffering DOS 2.1 & 2.2 user. Any computer buff who survived DOS 2.1, can survive anything....plague....plane crashes.... scuba tank out of air at depth....anything you can imagine. One of the NEATEST and most useful features of NEWDOS is the program's ability to LPRINT out the contents of the video display ANYTIME you simultaneously press the keyboard's JKL keys. The following program was written with Bryan Mumford's assistance. Now, those TRS-80 users without the blessings of NEWDOS and/or non-disk systems without this extremely useful feature, may now enjoy its use for myriad applications. It has been tested with non-disk Level II, DOS 2.1 & 2.2, and will even work with NEWDOS if the '123' keys are programmed for the LPRINT clue.

Why use it with NEWDOS which already has the 'JKL' LPRINT out video feature? A good question, Gridley. Answer: quite simply because this is a new and IMPROVED 'JKL' that allows the user to TERMINATE the function ANYTIME by merely hitting the keyboard's spacebar, whereas plain 'ole NEWDOS 'JKL' continues printing out all 16 lines of video till done or the printer turned off.....which may or may not terminate the function. Thank Bryan Mumford, for this excellent feature that also may be used to upgrade NEWDOS+ if desired.

PROGRAM LOGIC AND FLOW:

Page 113 is the source code with comments and page 114 the object code for this subroutine.

If the program is loaded via DOS, it is self-initializing and needs no further SYSTEM, etc., commands. If loaded from cassette, you must go through the standard SYSTEM, name, ENTER then after loaded, BREAK, routines. We like the program name MUM since Bryan Mumford did most of the work on this program and besides, it is a QUIET yet resident program till called.

Lines 160 - 180 very efficiently LOAD the program's starting address of 32000 decimal into the video display control block's driver address at 401EH and 401FH in MEM. Lines 210 and 220 EXCHANGE alternate register pairs for AF, BC, DE, and HL registers so as not to mess up the program in progress. They are switched back again in lines 500 and 510 before jumping back to the standard video routine at 0458H in line 530.

Line 220 loads the value of the 'H I J K L M N O' keyboard row into A register = 28 decimal IF 'JKL' keys are simultaneously pressed (see page 29 Volume 1). Line 230 subtracts 28 from A register and line 240 GOTO's RETURN if NOT ZERO; i.e., 'JKL' are NOT pressed.

Line 250 loads the HL register pair with 3C00H =15360 decimal, the FIRST video MEM location of the 1024 video MEM bytes. Line 260 is our line counter with the E register intialized at the video display's 16 lines. Line 270 performs the same function as 260 except for the characters per line by intializing the C register at 64 and setting up the LABEL 'LOOP1' so that the program reinitializes C register at 64 after a carriage return IF it has not finished printing all 16 or IF you have not hit the spacebar.

Line 280 has the LABEL 'LOOP2' which the program will use in line 360 IF all 64 characters in a line have NOT been printed. CALL - TEST sends the program off to TEST to see if the printer's READY 'handshake' in lines 420 - 440 is present. Line 290 loads the A register with the NEXT character in video MEM to be LPRINTed and line 300 LPRINTs it. Lines 310 - 330 check the keyboard's spacebar MEM location at 14400 to see if you are pressing the spacebar, and IF you are, take a quick exit OUT from this subroutine.

Line 340 adds +1 to the HL video MEM counter and line 350 subtracts 1 from C register, the video characters per line counter and simultaneously (almost) sets the 'Z' flag. Now is a good time to remember that the Z-80 ONLY sets the flags in the flag register when INDIVIDUAL registers or MEM locations pointed to by (HL), (IX), & (IY) are either INCremented or DECremented and DOES NOT set the flags when any of the REGISTER PAIRS are INC or DEC. This is a small price to pay for limited compatibility with Dr. Faggin's earlier 8080 µP.

Line 370 TESTs printer ready IF line 360 fell through and lines 380 and 390 issue a carriage return before line 400 subtracts 1 from the line counter and line 410 sends us back for the next line IF the E register NOT ZERO. If it is ZERO, then line 420 makes a quick exit. Lines 470 - 520 are our quick exit with a carriage return for the printer, restore the original AF, BC, DE, & HL registers via EX AF,AF'& EXX, and then JumPs back to standard video at 0458H.

If you wish to use the '123' modification to NEWDOS+ so as to be able to hit 'spacebar' to stop printing, the changes for lines 220 and 230 are given on the bottom of page 113.

This 'spacebar' return to normal program is especially useful with duplex telephone MODEMs, Morse code, or radio teletype programs when and if one wishes a very FAST 'resume regular program'; i.e., such as during amateur radio contests when you are scanning the band between contacts while the line printer is chomping out the logbook on the last contact IF you did not store it in MEM as per Chapters 7 or 8.

EXTRA CREDIT QUESTION:
If you did not press 'JKL' (OR '123'), how many extra micro-seconds does this subroutine add to program execution time?
ANSWER: Look it up in back of Editor/Assembler Handbook and correct execution time for 1.77 MHz clock.

```
00100 ; 'JKL' LPRINT OUT VIDEO DISPLAY FOR NON-NEWDOS SYSTEMS
00110
00120 ;  NON-DISK LEVEL II OR DOS 2.1 & 2.2 OR UPGRADE NEWDOS+
00130
00140 ;  BY:  BRYAN MUMFORD AND BOB RICHARDSON - AUGUST 1979
00150
00160 W4UCH   EQU     7D00H           ;= 32000 DECIMAL
00170         ORG     401EH           ;LOAD 7D00H INTO VIDEO-
00180         DEFW    W4UCH           ;BLOCK DRIVER ADDRESS
00190         ORG     W4UCH           ;START HERE
00200         EX      AF,AF'          ;SWAP ALTERNATE REGISTER
00210         EXX                     ;SWAP ALTERNATE REGISTERS
00220         LD      A,(14338)       ;JKL KEYBOARD ROW
00230         CP      28              ;JKL PRESSED = 28 DECIMAL
00240         JR      NZ,RETURN       ;RETURN NORM VIDEO NOT 0
00250         LD      HL,3C00H        ;1ST VIDEO CHARACTER MEM
00260         LD      DE,16           ;NUMBER OF VIDEO LINES
00270 LOOP1   LD      BC,64           ;VIDEO CHARACTERS/LINE
00280 LOOP2   CALL    TEST            ;CHECK FOR PRINTER READY?
00290         LD      A,(HL)          ;1ST VIDEO CHARACTER TO A
00300         LD      (37E8H),A       ;OUTPUT CHAR. TO PRINTER
00310         LD      A,(14400)       ;KYBD LINE WITH SPACEBAR
00320         CP      128             ;SPACEBAR PRESSED = 128
00330         JR      Z,OUT           ;GOTO 'OUT' IF ZERO
00340         INC     HL              ;ADD +1 TO VIDEO LOCATION
00350         DEC     C               ;MINUS 1 TO CHAR. COUNTER
00360         JR      NZ,LOOP2        ;GOTO NEXT CHAR. NOT ZERO
00370         CALL    TEST            ;CHECK FOR PRINTER READY?
00380         LD      A,0DH           ;0DH = CARRIAGE RETURN
00390         LD      (37E8H),A       ;DO IT !
00400         DEC     E               ;MINUS 1 TO LINE COUNTER
00410         JR      NZ,LOOP1        ;START ON NEXT LINE NOT 0
00420         JR      OUT             ;QUICK EXIT IF DONE
00430 TEST    LD      A,(37E8H)       ;PRINTER READY = 63 = ?
00440         CP      63              ;SUBTRACT 63
00450         JR      NZ,TEST         ;LOOP TILL PRINTER READY
00460         RET                     ;RETURN - LINE AFTER CALL
00470 OUT     CALL    TEST            ;IS PRINTER READY ?
00480         LD      A,0DH           ;0DH = CARRIAGE RETURN
00490         LD      (37E8H),A       ;DO IT !
00500 RETURN  EX      AF,AF'          ;RETURN ORIG. REGISTER
00510         EXX                     ;RETURN ORIG. REGISTERS
00520         JP      0458H           ;GOTO STD. VIDEO ROUTINE
00530         END     W4UCH           ;EL FIN = EL BEGIN
```

* AFTERTHOUGHT: LINES 260 AND 270 SHOULD READ: LD E,16 AND LD C,64 RESPECTIVELY FOR BETTER PROGRAMMING PRACTICE.

NOTE:

With NEWDOS+ use keyboard's '123' to avoid 'JKL' confusion. Change lines 220 & 230 as follows. Spacebar will quick exit.

```
00220       LD      A,(14352)       ;KEYBOARD '01234567' ROW
00230       CP      14              ;123 PRESSED = 14 DECIMAL
```

```
7D00            00160 W4UCH   EQU     7D00H
401E            00170         ORG     401EH
401E 007D       00180         DEFW    W4UCH
7D00            00190         ORG     W4UCH
7D00 08         00200         EX      AF,AF'
7D01 D9         00210         EXX
7D02 3A0238     00220         LD      A,(14338)
7D05 FE1C       00230         CP      28
7D07 2038       00240         JR      NZ,RETURN
7D09 21003C     00250         LD      HL,3C00H
7D0C 111000     00260         LD      DE,16
7D0F 014000     00270 LOOP1   LD      BC,64
7D12 CD317D     00280 LOOP2   CALL    TEST
7D15 7E         00290         LD      A,(HL)
7D16 32E837     00300         LD      (37E8H),A
7D19 3A4038     00310         LD      A,(14400)
7D1C FE80       00320         CP      128
7D1E 2819       00330         JR      Z,OUT
7D20 23         00340         INC     HL
7D21 0D         00350         DEC     C
7D22 20EE       00360         JR      NZ,LOOP2
7D24 CD317D     00370         CALL    TEST
7D27 3E0D       00380         LD      A,0DH
7D29 32E837     00390         LD      (37E8H),A
7D2C 1D         00400         DEC     E
7D2D 20E0       00410         JR      NZ,LOOP1
7D2F 1808       00420         JR      OUT
7D31 3AE837     00430 TEST    LD      A,(37E8H)
7D34 FE3F       00440         CP      63
7D36 20F9       00450         JR      NZ,TEST
7D38 C9         00460         RET
7D39 CD317D     00470 OUT     CALL    TEST
7D3C 3E0D       00480         LD      A,0DH
7D3E 32E837     00490         LD      (37E8H),A
7D41 08         00500 RETURN  EX      AF,AF'
7D42 D9         00510         EXX
7D43 C35804     00520         JP      0458H
7D00            00530         END     W4UCH
00000 TOTAL ERRORS

LOOP1   7D0F 00270   00410
LOOP2   7D12 00280   00360
OUT     7D39 00470   00330 00420
RETURN  7D41 00500   00240
TEST    7D31 00430   00280 00370 00450 00470
W4UCH   7D00 00160   00180 00190 00530
```

- OBJECT CODE -

NOTE:

THIS PAGE PRINTED OUT USING NEWDOS+ AND '123' TO LPRINT VIDEO WITH 'SPACEBAR' TO STOP LPRINTING.

- CHAPTER 7 -

STORING VIDEO IN MEM FOR LATER RECALL
(notes from a lecture)

SYNOPSIS:
Here is an interesting exercise that will allow the user to store 5 complete video display frames in MEM for later recall. Storage is called by pressing the '123' keys simultaneously, and recall by pressing '456'. The 115 byte assembly language program and MEM storage only require 5235 bytes total so it will operate easily with any 16K MEM system. It operates equally well with non-disk Level II, DOS 2.1, DOS 2.2, DOS 2.3, and NEWDOS+. The program may be entered in about 5 minutes (without comments) using the R/S Editor/Assembler.

INTRODUCTION:
There are many occasions when TRS-80 programmers would like to store data from the video display to RAM memory for later recall and review, thus allowing selective 'JKL' LPRINT of that data desired for permanent hard copy record. Other applications include duplex telephone line MODEM operations where incoming data may be selectively stored in MEM for leisurely recall, as well as Morse code and/or radio teletype systems, etc.

This program is self-executing when loaded in the DOS mode; i.e., it will operate in DOS or disk BASIC with no further SYSTEM and /27000 commands. With non-disk Level II it is loaded using standard cassette procedures.

On the inside rear cover is a photo of the author's TRS-80 layout. Left to right: Western I/O IBM #2970 Selectric printer terminal, TRS-80 with 64K MEM, upper/lower case switch, Mumford Micro 3 speed clock, and dual cassettes. Lower right: dual Wangco 82 mini-floppy disks from Percom. The matching recessed work tables for the printer and TRS-80 are courtesy of Radio Shack.

PROGRAM LOGIC AND FLOW:
Since the "JKL" keys when pressed simultaneously are the NEWDOS+ clue to LPRINT the video display, we will use the "123" keyboard keys to tell our assembly language program when to store a complete video display frame, and the keyboard "456" keys to tell our program when to recall a frame. Following W6OVP, Dave Lien's advice to "KISS" (keep-it-simple-stupid), we will limit this demonstration program to storing 5 video display frames; i.e., 1024 X 5 = 5120 bytes. Further simplification includes:

1. Automatic sequencing of the stored video display frames.
2. If a 6th video frame is stored it will "wipe-out" the previous #1. If a 7th frame stored, #2, etc.. etc.
3. Recall also includes automatic sequencing.
4. The number of the stored frame is displayed in the lower right hand corner of the video display.

Page 117 is the source code and page 118 the object code of this program. This demonstration program is located at 27000

decimal so that it will work with any Level II 16K MEM system, whether non-disk or otherwise. It may be relocated in MEM anywhere you wish by changing lines 120, 130, 150, 160, and 170 appropriately.

Let's run through the program briefly, even though the comments are largely self-explanatory. The label COUNT is the 1 byte MEM location where the number of frames stored, 1 to 5, is stashed. The label PLACE is the 2 byte MEM location containing the address in MEM where each 1024 byte video display frame is stored. Lines 170-190 are a straightforward way of poking the START address into the video display control block's driver address at 401EH and 401FH, thus making the program self-executing when loaded via DOS. Lines 220 &'230 switch alternate Z-80 register pairs so as not to foul-up any machinations that may be going on in either your BASIC or assembly language program that may be running. Remember, the alternate register pairs AF', BC', DE', and HL' are never used by Level II ROM. Additionally, the EX and EXX opcodes take 4+ bytes less MEM than PUSH and POP opcodes. Lines 230-250 are optional and serve only to remind the user "how many frames" have been stored or "which frame" is being recalled. Lines 260-300 are the real 'work horses' of the program that quite simply test the keyboard's numbers row to see if either '123' or '456' keys are pressed. Lines 310-330 restore the original register pairs and then jumps back to the normal video display routine at 0458H. The entire subroutine, so far, requires only about 24 microseconds, so be assured it will not disrupt most any program you are running.

STORE moves the 1024 byte video display to PLACE whenever it is called, resets COUNT to 1 if 5 frames have been stored, updates PLACE by +1024, updates COUNT by +1, and then returns to lines 310-330 for a normal exit back to your program. RECALL is virtually identical to STORE except that now we will reverse the procedure and move the stored 1024 bytes from PLACE to the video display before returning to exit lines 310-330. RESET simply reinitializes COUNT to +1 and PLACE to 27200 decimal, which is where we started. If you have enough memory to store 10 to 20 video display frames it is very easy to modify the program to do so. All that is necessary is to change lines 390 and 540 to: "CP 20", if you wish to store and/or recall 20 video display frames. If you chose to leave the frame number display, lines 230-250, in the program it would display frame 10 = : 11 = ; 12 = < 13 = '=' 14 = > 15 = ? 16 = @ 17 = A 18 = B 19 = C 20 = D, unless modified.

PROGRAM SUMMARY:
The program logic is straightforward and its primary purpose is to accustom the user to moving video display data around as desired. It is the predecessor to the next Chapter, "Split Screen Video Display," which provides the user with two totally independent video displays with individual scrolling. Semi-automatic storage and recall of the RECEIVE segment of the video display is provided, but may be modified with what you learned in this Chapter to provide the same functions for the TRANSMIT sector if desired.

```
00100 ;STORING VIDEO IN RAM DEMONSTRATION PROGRAM - SV8
00110
00120         ORG     6978H           ;= 27000 DECIMAL
00130 COUNT   EQU     6978H           ;PROGRAM SAVES SPACE HERE
00140         DEFB    1               ;INITIALIZE BYTE AT + 1
00150 PLACE   EQU     6979H           ;PROGRAM SAVES SPACE HERE
00160         DEFW    27200           ;BEGIN STORAGE 27200 MEM
00170 START   EQU     27003           ;PROGRAM WILL CAREFULLY-
00180         ORG     401EH           ;POKE 27003 AT 401EH AND-
00190         DEFW    START           ;401FH VIDEO DISPLAY -
00200         ORG     START           ;CONTROL BLOCK.
00210         EX      AF,AF'          ;EXCHANGE ALT. REGISTERS
00220         EXX                     ;EXCHANGE BC,DE,HL REGS
00230         LD      A,(COUNT)       ;NO. OF FRAMES STORED
00240         ADD     A,48            ;CHANGE TO ASCII NUMBER
00250         LD      (16382),A       ;DISPLAY LOWER RT. CORNER
00260         LD      A,(14352)       ;KYBD 01234567 NOS. ROW
00270         CP      14              ;SUB 14 = '123' PRESSED
00280         JR      Z,STORE         ;GOTO 'STORE' IF ZERO
00290         CP      112             ;SUB 112 = '456' PRESSED
00300         JR      Z,RECALL        ;GOTO 'RECALL'IF ZERO
00310 ELFIN   EX      AF,AF'          ;RESTORE AF REGISTER
00320         EXX                     ;RESTORE BC, DE, HL REGS.
00330         JP      0458H           ;GOTO NORMAL VIDEO
00340 STORE   LD      HL,15360        ;BEGINNING VIDEO MEMORY
00350         LD      DE,(PLACE)      ;LAST HI MEM STORE
00360         LD      BC,1024         ;BYTES VIDEO TO MOVE
00370         LDIR                    ;MOVE THEM !
00380         LD      A,(COUNT)       ;NO. OF FRAMES STORED
00390         CP      5               ;SUBTRACT 5
00400         JR      Z,RESET         ;GOTO RESET IF ZERO
00410         LD      HL,(PLACE)      ;LAST HI MEM STORE
00420         LD      DE,1024         ;NO. OF BYTES MOVED
00430         ADD     HL,DE           ;ADD THEM UP
00440         LD      (PLACE),HL      ;UPDATE STORE LOCATION
00450         LD      A,(COUNT)       ;NO. OF FRAMES STORED
00460         INC     A               ;ADD +1
00470         LD      (COUNT),A       ;UPDATE FRAMES STORED
00480         JR      ELFIN           ;GOTO THE END
00490 RECALL  LD      HL,(PLACE)      ;MEM STORE LOCATION
00500         LD      DE,15360        ;1ST LINE VIDEO LOCATION
00510         LD      BC,1024         ;NUMBER OF BYTES TO MOVE
00520         LDIR                    ;DO IT
00530         LD      A,(COUNT)       ;NUMBER OF STORED FRAME
00540         CP      5               ;SUBTRACT 5
00550         JR      Z,RESET         ;GOTO RESET IF ZERO
00560         LD      A,(COUNT)       ;NO. OF STORED FRAME
00570         INC     A               ;ADD + 1 TO 'A' REGISTER
00580         LD      (COUNT),A       ;UPDATE FRAME COUNTER
00590         LD      HL,(PLACE)      ;CURRENT FRAME LOCATION
00600         LD      DE,1024         ;BYTES NOW DISPLAYED
00610         ADD     HL,DE           ;ADD + 1024 TO HL REG.
00620         LD      (PLACE),HL      ;UPDATE PLACE LOCATION
00630         JR      ELFIN           ;GOTO THE END
00640 RESET   LD      A,1             ;REINITIALIZE AT + 1
00650         LD      (COUNT),A       ;LOAD + 1 IN COUNT MEM
00660         LD      HL,27200        ;REINITIALIZED AT 27200
00670         LD      (PLACE),HL      ;STASH 27200 IN PLACE MEM
00680         JR      ELFIN           ;GOTO THE END
00690         END     COUNT           ;EL FIN = EL BEGIN
```

```
6978            00120           ORG     6978H
6978            00130 COUNT     EQU     6978H
6978 01         00140           DEFB    1
6979            00150 PLACE     EQU     6979H
6979 406A       00160           DEFW    27200
697B            00170 START     EQU     27003
401E            00180           ORG     401EH
401E 7B69       00190           DEFW    START
697B            00200           ORG     START
697B 08         00210           EX      AF,AF'
697C D9         00220           EXX
697D 3A7869     00230           LD      A,(COUNT)
6980 C630       00240           ADD     A,48
6982 32FE3F     00250           LD      (16382),A
6985 3A1038     00260           LD      A,(14352)
6988 FE0E       00270           CP      14
698A 2809       00280           JR      Z,STORE
698C FE70       00290           CP      112
698E 282B       00300           JR      Z,RECALL
6990 08         00310 ELFIN     EX      AF,AF'
6991 D9         00320           EXX
6992 C35804     00330           JP      0458H
6995 21003C     00340 STORE     LD      HL,15360
6998 ED5B7969   00350           LD      DE,(PLACE)
699C 010004     00360           LD      BC,1024
699F EDB0       00370           LDIR
69A1 3A7869     00380           LD      A,(COUNT)
69A4 FE05       00390           CP      5
69A6 2838       00400           JR      Z,RESET
69A8 2A7969     00410           LD      HL,(PLACE)
69AB 110004     00420           LD      DE,1024
69AE 19         00430           ADD     HL,DE
69AF 227969     00440           LD      (PLACE),HL
69B2 3A7869     00450           LD      A,(COUNT)
69B5 3C         00460           INC     A
69B6 327869     00470           LD      (COUNT),A
69B9 18D5       00480           JR      ELFIN
69BB 2A7969     00490 RECALL    LD      HL,(PLACE)
69BE 11003C     00500           LD      DE,15360
69C1 010004     00510           LD      BC,1024
69C4 EDB0       00520           LDIR
69C6 3A7869     00530           LD      A,(COUNT)
69C9 FE05       00540           CP      5
69CB 2813       00550           JR      Z,RESET
69CD 3A7869     00560           LD      A,(COUNT)
69D0 3C         00570           INC     A
69D1 327869     00580           LD      (COUNT),A
69D4 2A7969     00590           LD      HL,(PLACE)
69D7 110004     00600           LD      DE,1024
69DA 19         00610           ADD     HL,DE
69DB 227969     00620           LD      (PLACE),HL
69DE 18B0       00630           JR      ELFIN
69E0 3E01       00640 RESET     LD      A,1
69E2 327869     00650           LD      (COUNT),A
69E5 21406A     00660           LD      HL,27200
69E8 227969     00670           LD      (PLACE),HL
69EB 18A3       00680           JR      ELFIN
6978            00690           END     COUNT
00000 TOTAL ERRORS
```

- CHAPTER 8 -

SPLIT SREEN SEPARATE VIDEO ON ONE DISPLAY

Here is an interesting assembly language program that you may use in any application where completely independent dual video displays are an operating convenience or necessity. Typical applications include: duplex MODEM operation between 2 TRS-80s over a telephone line, radio teletype with fast break-in, and a TRS-80 Morse code transmit-receive system that would allow a "type ahead" capability using interrupts, but to name a few.

With the exception of a few Level II ROM CALLs and the video memory locations, the program will work with any micro-computer utilizing the Z-80 or 8080 microprocessors with only minor modifications. Though it is intended fundamentally as a teaching program to acquaint the user with techniques of manipulating video memory, its applications are only limited by the imagination of the assembly language programmer and go far beyond the obvious duplex Modem, radio teletype, and Morse code applications.

Figure 8-1 depicts the TRS-80 video display immediately after the program has been loaded. Note that the cursor is in the TRANSMIT sector. The cursor character we chose to use is identical to that used in the standard TRS-80 Basic program. It may be any variety you wish and is selected in line 175 of the program.

Figure 8-2 illustrates the video display after a few automatic CQ's have been transmitted. The cursor has been moved to the receive sector by pressing the CLEAR key which acts as the T/R (transmit/receive) switch; (see Chapter 10 for the author's "TRS-80 Morse Code Transmit & Receive Program," as it also uses the CLEAR key as the T/R switch). Since this Chapter is about "split-screen" video we will SIMULATE the separate inputs, transmit and receive, to the program by using the keyboard for generating output to the "A" register in both TRANSMIT and RECEIVE. It could just as well be a Morse code, radio teletype, or phone line MODEM program generating the outputs to the "A" register selected by the operator, or conversely, the program could be written to automatically 'toggle' back and forth between TRANSMIT and RECEIVE whenever the solitary letter 'K' (over to you) was received or transmitted.

Figures 8-3 and 8-4 illustrate a typical Morse code contact between the author's amateur radio station, W4UCH/2, and his old friend's station, W2CIX/4, operated by Dr. Bill Laird at the Florida Institue of Technology in Melbourne, Florida. Both stations utilize the W4UCH TRS-80 Morse Code Transmit and Receive Program.

Figure 8-5 illustrates the video display after the ENTER key has been pressed. Note that W2CIX's receive sector has been completely cleared (CLS RECEIVE sector only), and that the

DO NOT SKIP Vol. 1 @ $10 ppd.

VOLUME 3 - DISASSEMBLED HANDBOOK FOR TRS-80

- WILL BE PUBLISHED JULY/AUGUST '80 -

STATUS:
It is about 1/3rd complete as of May/June '80. The table of contents and Chapter headings have been finalized. This Volume will deal primarily with communications oriented applications of the TRS-80 (which is the author's bag) using assembly language programs and even some BASIC pgms. to illustrate points.

The readers will be presented with a number of programs that offer the choice of software versus hardware solutions to many communications related applications including:

- Time sharing (SPOOLING) demonstration programs
- UART serial transmit demonstration programs
- UART serial receive demonstration programs
- WOULD YOU BELIEVE? A homebrew $15 telephone line MODEM
- COMPUTER BULLETIN BOARD access software program
- ASCII ham radio transmit and receive program 110/300 baud
- Using TRS-80 interrupts demonstration programs
- A/D homebrew converter and demonstration programs
- D/A homebrew converter and demonstration programs
- And MUCH, MUCH more

WANT TO RESERVE A COPY? SEND NO MONEY! CLIP & SEND COUPON:

- -

YES, BILL ME $15. WHEN READY. WHEN YOU RECEIVE MY CHECK MAIL VOLUME 3 TO ME POSTPAID. _____ PLEASE CHECK IF DESIRED.

YES, SEND ME VOLUME 3 WHEN READY. SEND IT TO ME COD, @ $15. PLUS COD AND MAILING COSTS. _____ PLEASE CHECK IF DESIRED.

NAME :_____

STREET :_____

CITY :_____ STATE_____ ZIP_____

NOTE:
Volume 3 MAY list at considerably MORE than $15. depending on inflation. YOU are guaranteed MAXIMUM price of $15. IF you order now.

RICHCRAFT ENGINEERING LTD., DRAWER 1065
#1 WAHMEDA INDUSTRIAL PARK, CHAUTAUQUA, NEW YORK 14722

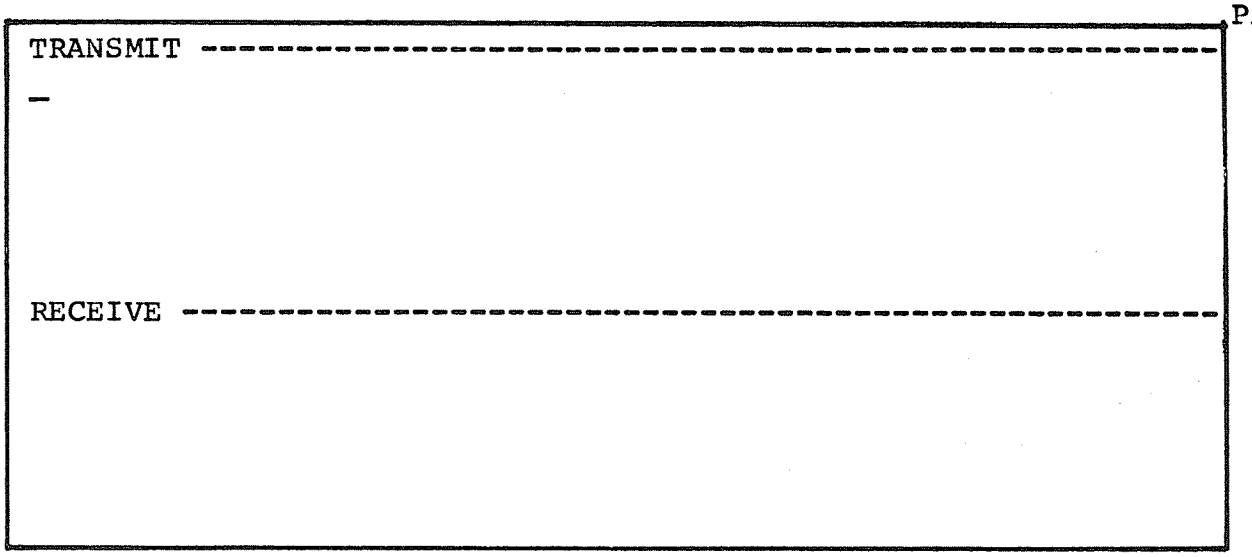


FIGURE 8-1

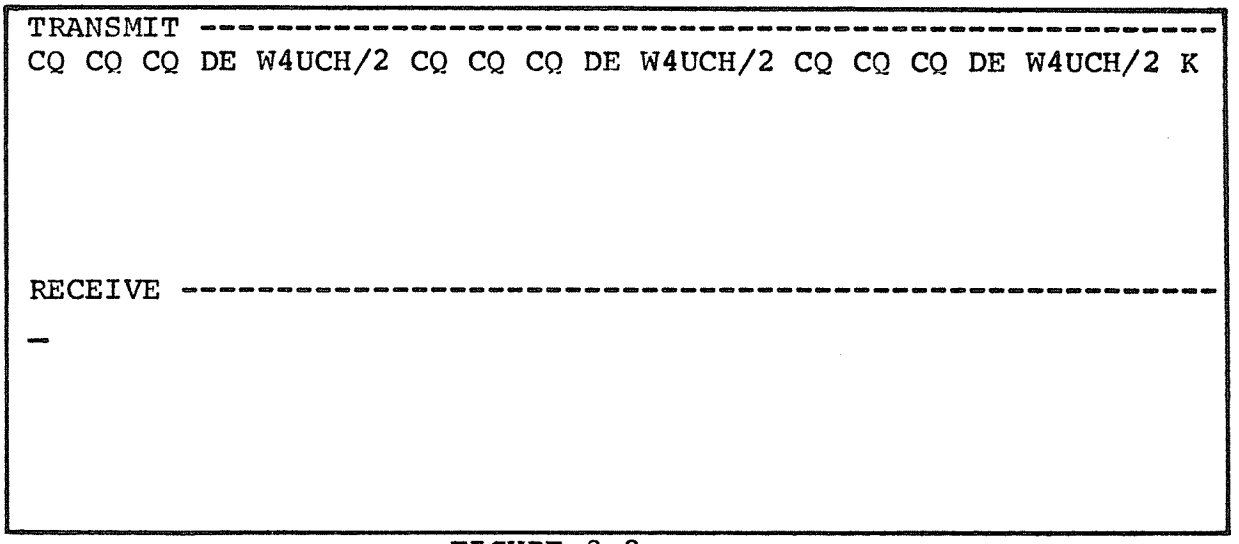


FIGURE 8-2

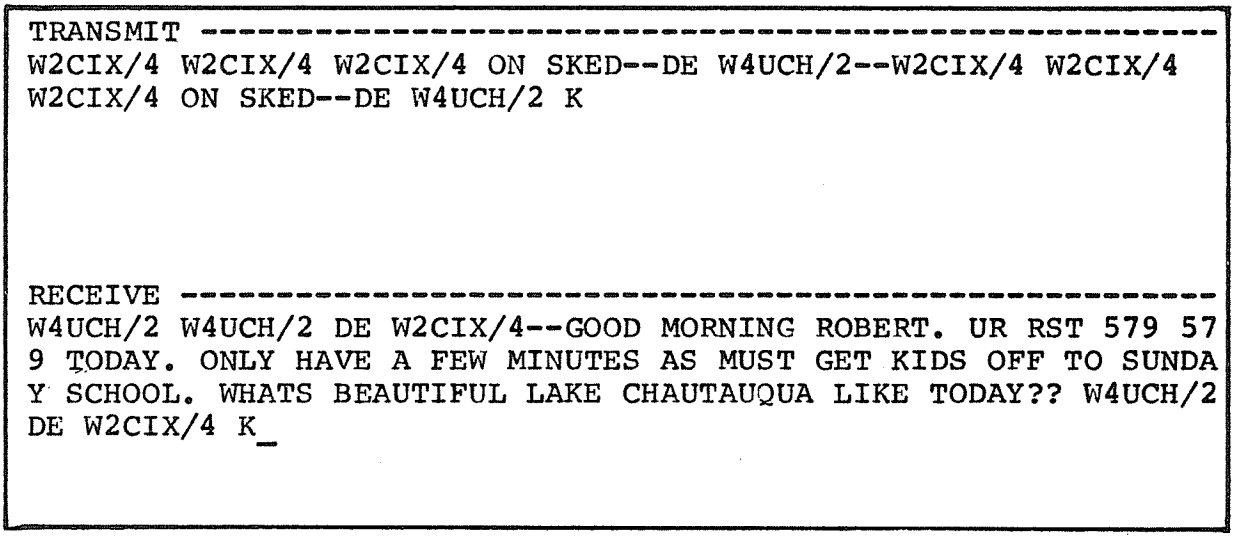


FIGURE 8-3

```
TRANSMIT ----------------------------------------------------
W2CIX/4 W2CIX/4 W2CIX/4 ON SKED--DE W4UCH/2--W2CIX/4 W2CIX/4
W2CIX/4 ON SKED--DE W4UCH/2 K BK BK W2CIX/4 DE W4UCH/4--HELLO
WILLIAM MATHEW--GUD TO HR U IN THERE--UR RST 559 559 BUT LOADS
OF QRM THIS MORNING--IT MUST BE A CONTEST WEEK?? LAKE FINALLY
FROZEN THIS WEEK--ABOUT 2 FEET SNO ON GND W2CIX/4 DE W4UCH/2 K

-

RECEIVE -----------------------------------------------------
W4UCH/2 W4UCH/2 DE W2CIX/4--GOOD MORNING ROBERT. UR RST 579 57
9 TODAY. ONLY HAVE A FEW MINUTES AS MUST GET KIDS OFF TO SUNDA
Y SCHOOL. WHATS BEAUTIFUL LAKE CHAUTAUQUA LIKE TODAY?? W4UCH/2
DE W2CIX/4 K W4UCH/2 DE W2CIX/4..OK ON WX THERE..UGH..IT IS 75
F HERE..SUNNY..THINK THR IS LOCAL CENTRAL FLA CW CONTEST ON TH
IS WEEKEND..FOR CD GANG..THEY DID MARVELOUS JOB WHEN TORNADOES
TOOK OUT PHONES NEAR ORLANDO LAST WEEK..W4UCH/2 DE W2CIX/4 K
```

FIGURE 8-4

```
TRANSMIT ---------------------------------------------------- 1
W2CIX/4 W2CIX/4 W2CIX/4 ON SKED--DE W4UCH/2--W2CIX/4 W2CIX/4
W2CIX/4 ON SKED--DE W4UCH/2 K BK BK W2CIX/4 DE W4UCH/4--HELLO
WILLIAM MATHEW--GUD TO HR U IN THERE--UR RST 559 559 BUT LOADS
OF QRM THIS MORNING--IT MUST BE A CONTEST WEEK?? LAKE FINALLY
FROZEN THIS WEEK--ABOUT 2 FEET SNO ON GND W2CIX/4 DE W4UCH/2 K


RECEIVE -----------------------------------------------------

-
```

FIGURE 8-5

```
TRANSMIT ---------------------------------------------------- 1









RECEIVE -----------------------------------------------------
W4UCH/2 W4UCH/2 DE W2CIX/4--GOOD MORNING ROBERT. UR RST 579 57
9 TODAY. ONLY HAVE A FEW MINUTES AS MUST GET KIDS OFF TO SUNDA
Y SCHOOL. WHATS BEAUTIFUL LAKE CHAUTAUQUA LIKE TODAY?? W4UCH/2
DE W2CIX/4 K W4UCH/2 DE W2CIX/4..OK ON WX THERE..UGH..IT IS 75
F HERE..SUNNY..THINK THR IS LOCAL CENTRAL FLA CW CONTEST ON TH
IS WEEKEND..FOR CD GANG..THEY DID MARVELOUS JOB WHEN TORNADOES
TOOK OUT PHONES NEAR ORLANDO LAST WEEK..W4UCH/2 DE W2CIX/4 K_
```

FIGURE 8-6

```
TRANSMIT ------------------------------------------------------ 1
?




RECEIVE -------------------------------------------------------
W4UCH/2 W4UCH/2 DE W2CIX/4--GOOD MORNING ROBERT. UR RST 579 57
9 TODAY. ONLY HAVE A FEW MINUTES AS MUST GET KIDS OFF TO SUNDA
Y SCHOOL. WHATS BEAUTIFUL LAKE CHAUTAUQUA LIKE TODAY?? W4UCH/2
DE W2CIX/4 K W4UCH/2 DE W2CIX/4..OK ON WX THERE..UGH..IT IS 75
F HERE..SUNNY..THINK THR IS LOCAL CENTRAL FLA CW CONTEST ON TH
IS WEEKEND..FOR CD GANG..THEY DID MARVELOUS JOB WHEN TORNADOES
TOOK OUT PHONES NEAR ORLANDO LAST WEEK..W4UCH/2 DE W2CIX/4 K
```

FIGURE 8-7

```
TRANSMIT ------------------------------------------------------ 2




RECEIVE -------------------------------------------------------
W4UCH/2 DE W2CIX/4 ITS THAT TIME OM..MUST RUN..SEE U NXT WK SA
ME TIME SAME FREQ..73 AND MY BEST TO NANCY..W4UCH/2 DE W2CIX/2
K K K_
```

FIGURE 8-8

LINE NO.	BEGINNING POSITION		END POSITION
1	15360	----------	15423
2	15424	----------	15487
3	15488	----------	15551
4	15552	----------	15615
5	15616	----------	15679
6	15680	----------	15743
7	15744	----------	15807
8	15808	----------	15871
9	15872	----------	15935
10	15936	----------	15999
11	16000	----------	16063
12	16064	----------	16127
13	16128	----------	16191
14	16192	----------	16255
15	16256	----------	16319
16	16320	----------	16383

FIGURE 8-9
TRS-80 Video Display - Memory Locations In Decimal

number '1' is now displayed on the right end of the TRANSMIT first line. This tells us that we have stored RECEIVE page 1 in high memory, and may now continue as desired. This sequence may go on and on depending upon how much memory you have available.

The 'up-arrow' key serves to CLS the TRANSMIT sector of the video display, and the 'down-arrow' key allows you to CLS the RECEIVE sector. If neither key is pressed after a long (511 bytes per sector) contact, either segment will automatically scroll 'up' one line from the bottom and CLS the bottom line when it is filled, similar to the standard video display.

Just like 2 separate TRS-80 vido displays? You bet it is, and that is just what the program was intended to do, plus allow you to optionally STORE in high memory up to 5 pages of the RECEIVE sector. Each page of the RECEIVE sector is configured at 511 bytes (1/2 the video display), with totally automatic sequencing after 5 pages are stored; i.e., the 6th is automatically stored in #1, and the original #1 is lost. If you wish, and you have adequate memory, this may be modified to allow storing up to 40 pages of RECEIVE sector in RAM memory, and/or stored automatically up to ANY number you wish on either disk or cassette, your choice. Another choice you have is to store both TRANSMIT and RECEIVE in RAM, disk or cassette memory. Whichever/whatever fulfills your needs may be used.

Figure 8-6 shows the video display after store #1 frame has been recalled from memory (with 'right-arrow' press number 1) and the TRANSMIT sector cleared with press 'up-arrow.'

Each time your press 'right-arrow,' a question mark '?' will appear in the first space of the TRANSMIT video display second line. This is asking you, "which stored page to recall?"

Figure 8-7 illustrates this function, and Figure 8-8 after the numeral '2' has been pressed to recall the second page we had stored in high memory. Please note that most Morse code contacts seldom have this perfect printout unless the code is generated by computer, as was this QSO (contact) at both ends. The W4UCH TRS-80 Morse program allows you to separately adjust dot and dash length to compensate for "swing-fisters" when receiving hand-key generated Morse code.

WHY ASSEMBLY LANGUAGE INSTEAD OF BASIC?

Let's take a look at this assembly language program in piece by piece fashion. Though the main program only occupies 584 bytes of memory, it appears quite lengthy to the inexperienced viewer. Had it been written in Basic, it would have required 5 to 10 times as many bytes to accomplish the same end. The most important aspect of the assembly language program with its resulting source code and object code, is the 'speed' with which it accomplishes its given task; i.e., approximately 250 to 350 times FASTER than the same program written in Basic.

In many programs speed is totally unimportant and in some programs determined solely by the input and/or output devices used which could be a keyboard and/or printer. In this case, speed is the primary criteria for 3 reasons:

1. We wish to be able to transmit as much information as quickly as the bandwidth available will allow.

2. We wish to use the Z-80 interrupt function to allow type-ahead operation while receiving. The TRANSMIT video memory will be used as a FIFO (first-in-first-out) buffer when we press the CLEAR key to change over from receive to transmit.

3. Since we are receiving and transmitting binary data, whether over a phone line or via Morse code/radio teletype, TIMING is the most important aspect used in coding and decoding each character. As such, if we wish to use an interrupt for type-ahead operation, we must do so VERY QUICKLY or our interrupt will completely "blow" the timing and our transmitted or received data will be meaningless garbage.

So much for all the good things that an assembly language program buys us when compared to one written in Basic. Let's get on with examining this specific program.

PROGRAM LOGIC AND FLOW:

Any program written in any language when broken down into small segments is simplesville personified and easily understood by a BRIGHT 6th grader. After all, the processor whether it be a 64 bits per byte 100 MHz Cray gizmo or our relatively sophisticated but small 8 bits per byte Z-80 running at 1.7 to 4.0 MHz, is doing nothing more than following the instructions the very human programmer gave it. A good programmer's software is invariably easy to follow and understand. A POOR programmer's software jumps around and is difficult to follow, no matter how complex or simple the software task may be. As we all know, there are many professional programmers today who would be much better off mixing concrete or carrying bricks at a construction site.

Let's see which category your author deserves by studying this split screen program in detail. Figure 9 depicts the TRS-80s 16 lines of video mem in decimal. Pages 125 - 128 are the program's source code with comments and pages 129 - 132 the object code.

```
00010 ; SPLIT-SCREEN DEMONSTRATION PROGRAM BY W4UCH/2
00015 ;
00020          ORG     7D00H           ;ORG = 32000 DECIMAL
00025 W4UCH    EQU     7D00H           ;T/R SIGNPOST HERE
00030          DEFW    0000H           ;T/R SIGNPOST HERE
00035          CALL    01C9H           ;CLS ROM SUBROUTINE
00040 RCURS    EQU     7D05H           ;RECEIVE CURSOR
00045          DEFW    15936           ;IN MEM LOCATION 7D05H
00050 VIDEND   EQU     7D07H           ;END OF 8/16TH VIDEO LINE
00055          DEFW    15871           ;STASH HERE IN 7D07H
00060 TCURS    EQU     7D09H           ;TRANSMIT CURSOR
00065          DEFW    15424           ;STASH HERE IN 7DO9H
00070 PLACE    EQU     7D0BH           ;HI-MEM STORAGE LOCATION
00075          DEFW    32768           ;INITIALIZE STORAGE HERE
00080 COUNT    EQU     7D0DH           ;NO. OF STORES 1 TO 5
00085          DEFB    00              ;INITIALIZE COUNT = 0
00090          LD      HL,32768        ;STORE MEM BEGINNING
00095          LD      DE,32769        ;PLUS ONE
00100          LD      BC,2555         ;BYTES TO CLEAR
00105          LD      (HL),32         ;CLEAR WITH ASCII SPACE
00110          LDIR                    ;CLEAR MEM FOR STORAGE
00115          LD      HL,XMIT         ;XMIT STRING MEM ADDRESS
00120          CALL    28A7H           ;DISPLAY $ SUBROUTINE
00125 XMIT     DEFM    'TRANSMIT ----------------------------------
----------------------'
00130          DEFB    00              ;END OF MESSAGE DELIMITER
00135          LD      HL,15872        ;= BEGINNING OF 9TH LINE-
00140          LD      (4020H),HL      ;MOVE CURSOR TO 9TH LINE.
00145          LD      HL,RECV         ;STRING MEM ADDRESS
00150          CALL    28A7H           ;DISPLAY $ SUBROUTINE
00155 RECV     DEFM    'RECEIVE -----------------------------------
----------------------'
00160          DEFB    00              ;END OF MESSAGE DELIMITER
00165          LD      HL,15424        ;MOVE CURSOR TO 2ND LINE
00170          LD      (4020H),HL      ;DO IT
00175 KYBD     LD      A,95            ;CURSOR CHARACTER
00180          LD      HL,(4020H)      ;CURSOR POSITION TO HL
00185          LD      (HL),A          ;DISPLAY CURSOR
00190          CALL    0049H           ;INKEY$ TO A REGISTER
00195          CP      31              ;T/R 'CLEAR' KEY PRESSED?
00200          JR      Z,TRSW          ;IF 0 GOTO T/R SUBROUTINE
00205          CP      10              ;IS DOWN-ARROW PRESSED?
00210          JP      Z,RCLS          ;IF SO GOTO RECEIVE CLS
00215          CP      91              ;IS UP-ARROW PRESSED?
00220          JP      Z,TCLS          ;IF SO GOTO TRANSMIT CLS
00225          CP      13              ;IS IT 'ENTER' PRESSED?
00230          JP      Z,STORE         ;IF 0 STORE RECEIVE FRAME
00235          CP      9               ;IS RIGHT-ARROW PRESSED?
00240          JP      Z,RESTOR        ;IF SO RESTORE FM MEM
00245          CALL    032AH           ;DISPLAY A REG ON VIDEO
00250          LD      HL,(4020H)      ;CURSOR POS TO HL REG
00255          LD      DE,(VIDEND)     ;END OF 8/16TH VIDEO LINE
00260          CALL    1C90H           ;COMPARE HL-DE SET Z FLAG
00265          JR      Z,SCROLL        ;IF ZERO GOTO SCROLL
```

```
00270           JR      KYBD            ;BACK TO KEYBOARD
00275 SCROLL    LD      A,(W4UCH)       ;T/R SIGNPOST 0=T & 1=R
00280           CP      00              ;SUBTRACT ZERO
00285           JR      Z,TSCROL        ;IF 0 GOTO XMIT SCROLL
00290           LD      HL,16000        ;RECV SCRL BEGIN LINE 11
00295           LD      DE,15936        ;MOVE TO LINE 10 VIDEO
00300           LD      BC,384          ;NUMBER BYTES TO MOVE
00305           LDIR                    ;MOVE IT. DEC BC TO ZERO
00310           LD      HL,16320        ;BEGINNING LINE 16 VIDEO
00315           LD      DE,16321        ;PLUS ONE
00320           LD      BC,63           ;NUMBER OF BYTES TO MOVE
00325           LD      (HL),32         ;32 = ASCII SPACE
00330           LDIR                    ;MOVE IT. DEC BC TO ZERO
00335           LD      HL,16320        ;BEGIN 16TH VIDEO LINE
00340           LD      (4020H),HL      ;RESET CURSOR
00345           JR      KYBD            ;BACK TO KEYBOARD
00350 TSCROL    LD      HL,15488        ;BEGINNING 3RD LINE VIDEO
00355           LD      DE,15424        ;TO 2ND LINE OF VIDEO
00360           LD      BC,384          ;NUMBER OF BYTES TO MOVE
00365           LDIR                    ;MOVE IT. DEC BC TO ZERO
00370           LD      HL,15808        ;BEGINNING 8TH VIDEO LINE
00375           LD      DE,15809        ;PLUS ONE
00380           LD      BC,63           ;NUMBER OF BYTES TO MOVE
00385           LD      (HL),32         ;32 = ASCII SPACE
00390           LDIR                    ;DO IT. DEC BC TO ZERO
00395           LD      HL,15808        ;BEGIN 8TH VIDEO LINE
00400           LD      (4020H),HL      ;RESET CURSOR
00405           JR      KYBD            ;BACK TO KEYBOARD
00410 TRSW      LD      A,(W4UCH)       ;SIGNPOST MEM ADDRESS
00415           CP      0               ;SUBTRACT ZERO
00420           JR      Z,CHANGE        ;IF 0 GOTO CHANGE
00425           LD      A,0             ;T/R SET TO TRANSMIT
00430           LD      (W4UCH),A       ;STASH IT IN MEM
00435           LD      A,32            ;= ASCII SPACE
00440           LD      HL,(4020H)      ;CURSOR POSITION
00445           LD      (HL),A          ;REMOVE CURSOR
00450           LD      (RCURS),HL      ;SAVE RECV CURS POSITION
00455           LD      HL,(TCURS)      ;XMIT CURSOR POSITION
00460           LD      (4020H),HL      ;LOAD INTO VIDEO DCB
00465           LD      HL,15871        ;END OF 8TH VIDEO LINE
00470           LD      (VIDEND),HL     ;UPDATE FOR TRANSMIT
00475           JP      KYBD            ;RETURN TO KEYBOARD
00480 CHANGE    LD      A,1             ;T/R SET TO RECEIVE
00485           LD      (W4UCH),A       ;STASH IT IN MEM
00490           LD      A,32            ;= ASCII SPACE
00495           LD      HL,(4020H)      ;CURSOR POSITION
00500           LD      (HL),A          ;REMOVE CURSOR
00505           LD      (TCURS),HL      ;SAVE XMIT CURS POSITION
00510           LD      HL,(RCURS)      ;RECV CURSOR POSITION
00515           LD      (4020H),HL      ;LOAD INTO VIDEO DCB
00520           LD      HL,16383        ;END OF 16TH VIDEO LINE
00525           LD      (VIDEND),HL     ;UPDATE FOR RECEIVE
00530           JP      KYBD            ;RETURN TO KEYBOARD
00535 RCLS      LD      A,(W4UCH)       ;T/R SIGNPOST 0=T & 1=R
```

```
00540           CP      1               ;SUBTRACT 1
00545           JP      NZ,KYBD         ;GOTO KYBD IF NOT RECEIVE
00550           LD      HL,15936        ;BEGINNING 9TH VIDEO LINE
00555           LD      DE,15937        ;PLUS ONE
00560           LD      BC,446          ;NUMBER BYTES TO MOVE
00565           LD      (HL),32         ;= ASCII SPACE
00570           LDIR                    ;CLS RECEIVE VIDEO SECTOR
00575           LD      HL,15936        ;MOVE CURSOR TO 9TH LINE
00580           LD      (4020H),HL      ;DO IT
00585           JP      KYBD            ;GO BACK TO KEYBOARD
00590 TCLS      LD      A,(W4UCH)       ;T/R SIGNPOST 0=T & 1=R
00595           CP      0               ;SUBTRACT ZERO
00600           JP      NZ,KYBD         ;GOTO KYBD IF RECEIVE
00605           LD      HL,15424        ;BEGINNING 2ND LINE VIDEO
00610           LD      DE,15425        ;PLUS ONE
00615           LD      BC,446          ;NUMBER BYTES TO MOVE
00620           LD      (HL),32         ;= ASCII SPACE
00625           LDIR                    ;CLS TRANSMIT VIDEO SECT.
00630           LD      HL,15424        ;MOVE CURSOR TO 2ND LINE
00635           LD      (4020H),HL      ;DO IT
00640           JP      KYBD            ;GO BACK TO KYBD
00645 STORE     LD      A,(W4UCH)       ;T/R SIGNPOST
00650           CP      1               ;IS IT RECEIVE?
00655           JP      NZ,KYBD         ;IF NOT RECV GOTO KYBD
00660           LD      A,(COUNT)       ;NUMBER OF STORES IN MEM
00665           INC     A               ;ADD 1 TO "A" REGISTER
00670           CP      6               ;SUBTRACT 6
00675           JP      Z,OUT           ;GOTO OUT IF ZERO & RESET
00680           LD      (COUNT),A       ;UPDATE STORE COUNTER
00685           ADD     A,48            ;CONVERT TO ASCII NUMBER
00690           LD      (15423),A       ;DISPLAY - 1ST LINE VIDEO
00695           LD      HL,15872        ;BEGINNING RECEIVE VIDEO
00700           LD      DE,(PLACE)      ;MEM 'STORE' LOCATION
00705           LD      BC,511          ;NUMBER OF BYTES TO MOVE
00710           LDIR                    ;MOVE IT
00715           LD      HL,(PLACE)      ;LAST HI-MEM STORE PLACE
00720           LD      DE,511          ;FOR 511 BYTES
00725           ADD     HL,DE           ;ADD THEM UP
00730           LD      (PLACE),HL      ;PUT THEM IN PLACE MEM
00735           JP      RCLS            ;GOTO CLS RECEIVE SECT.
00740 RESTOR    LD      A,(W4UCH)       ;T/R SIGNPOST
00745           CP      0               ;SUBTRACT ZERO
00750           JP      NZ,KYBD         ;GOTO KYBD IF RECEIVE
00755           LD      A,63            ;= ASCII ?
00760           LD      (15424),A       ;DISPLAY IN 1ST XMIT LINE
00765           CALL    0049H           ;KYBD INPUT A REGISTER
00770           CP      49              ;IS IT ASCII ONE?
00775           JR      Z,ONE           ;GOTO ONE FRAME LOCATION
00780           CP      50              ;IS IT ASCII TWO?
00785           JR      Z,TWO           ;GOTO TWO FRAME LOCATION
00790           CP      51              ;IS IT ASCII THREE?
00795           JR      Z,THREE         ;GOTO THREE FRAME LOCAT.
00800           CP      52              ;IS IT ASCII FOUR?
```

```
00805           JR      Z,FOUR          ;GOTO FOUR FRAME LOCATION
00810           CP      53              ;IS IT ASCII FIVE?
00815           JR      Z,FIVE          ;GOTO FIVE FRAME LOCATION
00820           JP      KYBD            ;IF NONE GOTO KEYBOARD
00825 TELL      LD      HL,(SHOW)       ;STORE MEM LOCATION
00830           LD      DE,15872        ;RECV VIDEO ADDRESS
00835           LD      BC,511          ;NO. BYTES TO MOVE
00840           LDIR                    ;MOVE THEM
00845           JP      KYBD            ;RETURN TO KEYBOARD
00850 SHOW      DEFS    2               ;SAVE 2 BYTES
00855 ONE       LD      HL,32768        ;FIRST MEM STORE
00860           LD      (SHOW),HL       ;PUT IT IN SHOW MEMORY
00865           JR      TELL            ;GOTO TELL
00870 TWO       LD      HL,33279        ;SECOND MEM STORE
00875           LD      (SHOW),HL       ;PUT IT IN SHOW MEMORY
00880           JR      TELL            ;GOTO TELL
00885 THREE     LD      HL,33790        ;THIRD MEM STORE
00890           LD      (SHOW),HL       ;PUT IT IN SHOW MEMORY
00895           JR      TELL            ;GOTO TELL
00900 FOUR      LD      HL,34301        ;FOURTH MEM STORE
00905           LD      (SHOW),HL       ;PUT IT IN SHOW MEMORY
00910           JR      TELL            ;GOTO TELL
00915 FIVE      LD      HL,34812        ;FIFTH MEM STORE
00920           LD      (SHOW),HL       ;PUT IT IN SHOW MEMORY
00925           JR      TELL            ;GOTO TELL
00930 OUT       LD      A,0             ;ZERO TO "A" REGISTER
00935           LD      (COUNT),A       ;RESET STORE COUNTER 0
00940           LD      HL,32768        ;BEGINNING STORE MEMORY
00945           LD      (PLACE),HL      ;RESET STORE MEMORY
00950           JP      KYBD            ;RETURN TO KEYBOARD
00955           END     W4UCH           ;AMATEUR CALL LETTERS
00960 ;
00965 ; PROGRAM UTILIZES 584 BYTES PLUS 2555 BYTES STORAGE
```

- FIGURE 8-10 -

SOURCE CODE

```
7D00            00020           ORG     7D00H
7D00            00025 W4UCH     EQU     7D00H
7D00 0000       00030           DEFW    0000H
7D02 CDC901     00035           CALL    01C9H
7D05            00040 RCURS     EQU     7D05H
7D05 403E       00045           DEFW    15936
7D07            00050 VIDEND    EQU     7D07H
7D07 FF3D       00055           DEFW    15871
7D09            00060 TCURS     EQU     7D09H
7D09 403C       00065           DEFW    15424
7D0B            00070 PLACE     EQU     7D0BH
7D0B 0080       00075           DEFW    32768
7D0D            00080 COUNT     EQU     7D0DH
7D0D 00         00085           DEFB    00
7D0E 210080     00090           LD      HL,32768
7D11 110180     00095           LD      DE,32769
7D14 01FB09     00100           LD      BC,2555
7D17 3620       00105           LD      (HL),32
7D19 EDB0       00110           LDIR
7D1B 21217D     00115           LD      HL,XMIT
7D1E CDA728     00120           CALL    28A7H
7D21 54         00125 XMIT      DEFM    'TRANSMIT ---------
----------------------------------------'
7D61 00         00130           DEFB    00
7D62 21003E     00135           LD      HL,15872
7D65 222040     00140           LD      (4020H),HL
7D68 216E7D     00145           LD      HL,RECV
7D6B CDA728     00150           CALL    28A7H
7D6E 52         00155 RECV      DEFM    'RECEIVE ----------
----------------------------------------'
7DAE 00         00160           DEFB    00
7DAF 21403C     00165           LD      HL,15424
7DB2 222040     00170           LD      (4020H),HL
7DB5 3E5F       00175 KYBD      LD      A,95
7DB7 2A2040     00180           LD      HL,(4020H)
7DBA 77         00185           LD      (HL),A
7DBB CD4900     00190           CALL    0049H
7DBE FE1F       00195           CP      31
7DC0 286C       00200           JR      Z,TRSW
7DC2 FE0A       00205           CP      10
7DC4 CA6F7E     00210           JP      Z,RCLS
7DC7 FE5B       00215           CP      91
7DC9 CA8D7E     00220           JP      Z,TCLS
7DCC FE0D       00225           CP      13
7DCE CAAB7E     00230           JP      Z,STORE
7DD1 FE09       00235           CP      9
7DD3 CADD7E     00240           JP      Z,RESTOR
7DD6 CD2A03     00245           CALL    032AH
7DD9 2A2040     00250           LD      HL,(4020H)
7DDC ED5B077D   00255           LD      DE,(VIDEND)
7DE0 CD901C     00260           CALL    1C90H
7DE3 2802       00265           JR      Z,SCROLL
```

```
7DE5 18CE     00270          JR      KYBD
7DE7 3A007D   00275 SCROLL   LD      A,(W4UCH)
7DEA FE00     00280          CP      00
7DEC 2820     00285          JR      Z,TSCROL
7DEE 21803E   00290          LD      HL,16000
7DF1 11403E   00295          LD      DE,15936
7DF4 018001   00300          LD      BC,384
7DF7 EDB0     00305          LDIR
7DF9 21C03F   00310          LD      HL,16320
7DFC 11C13F   00315          LD      DE,16321
7DFF 013F00   00320          LD      BC,63
7E02 3620     00325          LD      (HL),32
7E04 EDB0     00330          LDIR
7E06 21C03F   00335          LD      HL,16320
7E09 222040   00340          LD      (4020H),HL
7E0C 18A7     00345          JR      KYBD
7E0E 21803C   00350 TSCROL   LD      HL,15488
7E11 11403C   00355          LD      DE,15424
7E14 018001   00360          LD      BC,384
7E17 EDB0     00365          LDIR
7E19 21C03D   00370          LD      HL,15808
7E1C 11C13D   00375          LD      DE,15809
7E1F 013F00   00380          LD      BC,63
7E22 3620     00385          LD      (HL),32
7E24 EDB0     00390          LDIR
7E26 21C03D   00395          LD      HL,15808
7E29 222040   00400          LD      (4020H),HL
7E2C 1887     00405          JR      KYBD
7E2E 3A007D   00410 TRSW     LD      A,(W4UCH)
7E31 FE00     00415          CP      0
7E33 281D     00420          JR      Z,CHANGE
7E35 3E00     00425          LD      A,0
7E37 32007D   00430          LD      (W4UCH),A
7E3A 3E20     00435          LD      A,32
7E3C 2A2040   00440          LD      HL,(4020H)
7E3F 77       00445          LD      (HL),A
7E40 22057D   00450          LD      (RCURS),HL
7E43 2A097D   00455          LD      HL,(TCURS)
7E46 222040   00460          LD      (4020H),HL
7E49 21FF3D   00465          LD      HL,15871
7E4C 22077D   00470          LD      (VIDEND),HL
7E4F C3B57D   00475          JP      KYBD
7E52 3E01     00480 CHANGE   LD      A,1
7E54 32007D   00485          LD      (W4UCH),A
7E57 3E20     00490          LD      A,32
7E59 2A2040   00495          LD      HL,(4020H)
7E5C 77       00500          LD      (HL),A
7E5D 22097D   00505          LD      (TCURS),HL
7E60 2A057D   00510          LD      HL,(RCURS)
7E63 222040   00515          LD      (4020H),HL
7E66 21FF3F   00520          LD      HL,16383
7E69 22077D   00525          LD      (VIDEND),HL
7E6C C3B57D   00530          JP      KYBD
7E6F 3A007D   00535 RCLS     LD      A,(W4UCH)
```

```
7E72 FE01      00540          CP      1
7E74 C2B57D    00545          JP      NZ,KYBD
7E77 21403E    00550          LD      HL,15936
7E7A 11413E    00555          LD      DE,15937
7E7D 01BE01    00560          LD      BC,446
7E80 3620      00565          LD      (HL),32
7E82 EDB0      00570          LDIR
7E84 21403E    00575          LD      HL,15936
7E87 222040    00580          LD      (4020H),HL
7E8A C3B57D    00585          JP      KYBD
7E8D 3A007D    00590 TCLS     LD      A,(W4UCH)
7E90 FE00      00595          CP      0
7E92 C2B57D    00600          JP      NZ,KYBD
7E95 21403C    00605          LD      HL,15424
7E98 11413C    00610          LD      DE,15425
7E9B 01BE01    00615          LD      BC,446
7E9E 3620      00620          LD      (HL),32
7EA0 EDB0      00625          LDIR
7EA2 21403C    00630          LD      HL,15424
7EA5 222040    00635          LD      (4020H),HL
7EA8 C3B57D    00640          JP      KYBD
7EAB 3A007D    00645 STORE    LD      A,(W4UCH)
7EAE FE01      00650          CP      1
7EB0 C2B57D    00655          JP      NZ,KYBD
7EB3 3A0D7D    00660          LD      A,(COUNT)
7EB6 3C        00665          INC     A
7EB7 FE06      00670          CP      6
7EB9 CA3C7F    00675          JP      Z,OUT
7EBC 320D7D    00680          LD      (COUNT),A
7EBF C630      00685          ADD     A,48
7EC1 323F3C    00690          LD      (15423),A
7EC4 21003E    00695          LD      HL,15872
7EC7 ED5B0B7D  00700          LD      DE,(PLACE)
7ECB 01FF01    00705          LD      BC,511
7ECE EDB0      00710          LDIR
7ED0 2A0B7D    00715          LD      HL,(PLACE)
7ED3 11FF01    00720          LD      DE,511
7ED6 19        00725          ADD     HL,DE
7ED7 220B7D    00730          LD      (PLACE),HL
7EDA C36F7E    00735          JP      RCLS
7EDD 3A007D    00740 RESTOR   LD      A,(W4UCH)
7EE0 FE00      00745          CP      0
7EE2 C2B57D    00750          JP      NZ,KYBD
7EE5 3E3F      00755          LD      A,63
7EE7 32403C    00760          LD      (15424),A
7EEA CD4900    00765          CALL    0049H
7EED FE31      00770          CP      49
7EEF 2823      00775          JR      Z,ONE
7EF1 FE32      00780          CP      50
7EF3 2827      00785          JR      Z,TWO
7EF5 FE33      00790          CP      51
7EF7 282B      00795          JR      Z,THREE
7EF9 FE34      00800          CP      52
```

```
          7EFB 282F     00805          JR      Z,FOUR
          7EFD FE35     00810          CP      53
          7EFF 2833     00815          JR      Z,FIVE
          7F01 C3B57D   00820          JP      KYBD
          7F04 2A127F   00825 TELL     LD      HL,(SHOW)
          7F07 11003E   00830          LD      DE,15872
          7F0A 01FF01   00835          LD      BC,511
          7F0D EDB0     00840          LDIR
          7F0F C3B57D   00845          JP      KYBD
          0002          00850 SHOW     DEFS    2
          7F14 210080   00855 ONE      LD      HL,32768
          7F17 22127F   00860          LD      (SHOW),HL
          7F1A 18E8     00865          JR      TELL
          7F1C 21FF81   00870 TWO      LD      HL,33279
          7F1F 22127F   00875          LD      (SHOW),HL
          7F22 18E0     00880          JR      TELL
          7F24 21FE83   00885 THREE    LD      HL,33790
          7F27 22127F   00890          LD      (SHOW),HL
          7F2A 18D8     00895          JR      TELL
          7F2C 21FD85   00900 FOUR     LD      HL,34301
          7F2F 22127F   00905          LD      (SHOW),HL
          7F32 18D0     00910          JR      TELL
          7F34 21FC87   00915 FIVE     LD      HL,34812
          7F37 22127F   00920          LD      (SHOW),HL
          7F3A 18C8     00925          JR      TELL
          7F3C 3E00     00930 OUT      LD      A,0
          7F3E 320D7D   00935          LD      (COUNT),A
          7F41 210080   00940          LD      HL,32768
          7F44 220B7D   00945          LD      (PLACE),HL
          7F47 C3B57D   00950          JP      KYBD
          7D00          00955          END     W4UCH
          00000 TOTAL ERRORS
CHANGE  7E52 00480   00420
COUNT   7D0D 00080   00660 00680 00935
FIVE    7F34 00915   00815
FOUR    7F2C 00900   00805
KYBD    7DB5 00175   00270 00345 00405 00475 00530 00545 00585
                     00600 00640 00655 00750 00820 00845 00950
ONE     7F14 00855   00775
OUT     7F3C 00930   00675
PLACE   7D0B 00070   00700 00715 00730 00945
RCLS    7E6F 00535   00210 00735
RCURS   7D05 00040   00450 00510
RECV    7D6E 00155   00145
RESTOR  7EDD 00740   00240
SCROLL  7DE7 00275   00265
SHOW    7F12 00850   00825 00860 00875 00890 00905 00920
STORE   7EAB 00645   00230
TCLS    7E8D 00590   00220
TCURS   7D09 00060   00455 00505
TELL    7F04 00825   00865 00880 00895 00910 00925
THREE   7F24 00885   00795
TRSW    7E2E 00410   00200
TSCROL  7E0E 00350   00285
TWO     7F1C 00870   00785
VIDEND  7D07 00050   00255 00470 00525
W4UCH   7D00 00025   00275 00410 00430 00485 00535 00590 00645
                     00740 00955
XMIT    7D21 00125   00115
```

The video MEM shown in Figure 9 consists of real RAM (random access memory) locations and may be used SIMILAR to (7 bits worth) any Level II memory location = to or above 17129 decimal in standard Level II and = to or above 26810 when using NEWDOS+ disk Basic. Figure 9's video memory locations are in decimal and are included as a reference since our assembly language program will use decimal for video. The excellent Radio Shack Editor/Assembler could care less whether we use decimal or hexadecimal, just as long as we remember to include an 'H' at the end of each hex number and a zero at the beginning IF the hex no. STARTS with a letter of the alphabet.

We think the program is much easier to follow with video locations in decimal, and as such will use them throughout. Purists who insist on everything in hex may skip ahead to the next Chapter as it is often said that anyone who visualizes all numbers in hexadecimal must of necessity 'run rabbits and bark at the moon.' We have enough to keep us busy without a pack of baying hounds around to complicate the otherwise simple matter at hand.

LET'S GET ON WITH THE PROGRAM:

The program may be broken down into six fundamental segments each with its own unique job to perform that is virtually independent of the others. The six segments are:

1. Initialization

2. Keyboard input/control (we use the keyboard as the data source for both TRANSMIT & RECEIVE in this demo program.)

3. Auto-scroll for both TRANSMIT & RECEIVE segments.

4. Separate CLS for both segments.

5. Semi-automatic storage of video RECEIVE memory.

6. Recall and display of stored RECEIVE segments.

This program begins at MEM location 32000 decimal for no particular reason whatsoever other than habit. If you have a Level II TRS-80 with only 4K MEM by all means locate it at 17129 decimal = 4E29 hex and you will have MEM to spare for the 5 pages of RECEIVE video storage as the main program uses 584 bytes + 2555 bytes storage = 3139 bytes total. If you have 16K MEM, 29000 decimal = 7148 hex would be a good location to begin the program.

INITIALIZATION SEGMENT:

Figure 10 is the source code and Figure 11 the object code. To save program MEM (it takes less bytes) there are 6 labels that must be redefined here if you choose to move the program. They are: W4UCH, RCURS, VIDEND, TCURS, PLACE, and COUNT. It is a simple matter to do so. Let's assume we have a Level II

TRS-80 with 16K MEM and wish to relocate the program origin to 29000 decimal = 7148H. The following 7 lines would be changed:

```
00020          ORG     7148H          ;ORG = 29000 DECIMAL
00025 W4UCH    EQU     7148H          ;T/R SIGNPOST HERE
00040 RCURS    EQU     714DH          ;RECEIVE CURSOR
00050 VIDEND   EQU     714FH          ;END OF 8/16TH VIDEO LINE
00060 TCURS    EQU     7151H          ;TRANSMIT CURSOR
00070 PLACE    EQU     7153H          ;INITIALIZE STORAGE HERE
00080 COUNT    EQU     7155H          ;NO. OF STORES 1 TO 5
```

Our mid-1977 TRS-80 does not particularly like relocatable code (it abhors it), so this is an additional reason for writing these lines in this format, besides the memory saving feature.

Even though Figure 10's source code includes comments, let us expand a bit (no pun intended) on the more important program lines and comments.

025: W4UCH MEM location holds the TRANSMIT/RECEIVE signpost byte. A zero = TRANSMIT and a 1 = RECEIVE. This MEM location could have been 1 byte rather than 2 bytes, DEFB instead of DEFW, but uses 2 for future expansion.

040: Is a 2 byte location to hold the RECEIVE cursor address.

050: Is a 2 byte location to hold the address of the end of the last video line in either TRANSMIT or RECEIVE, thus allowing both automatic scrolling and STORE in receive.

060: Same as 040 for the TRANSMIT cursor.

070: This MEM location holds the automatically sequenced address for storing the 1 to 5 RECEIVE frames.

080: COUNT holds the number, 1 to 5 of RECEIVE frames stored.

090: Through 110 clears storage memory to ASCII zero. "LINES 090 & 095 MUST BE CHANGED SHOULD YOU RELOCATE THE PGM."

115: Through 160 display the TRANSMIT and RECEIVE segments on the video display using the ROM display $ function.

KEYBOARD SEGMENT:

190: This is the Level II ROM subroutine similar to the Basic INKEY$ function when used in a loop. It loops until a character on the keyboard is pressed, the character placed in the "A" register, and then RETurns to the following line.

195: The compares and jump if zeros from lines 195 through 240 direct the program off to the appropriate subroutine called from the keyboard. Let's look at line no. 245 before chasing off and tracking down each subroutine.

245: CALL 032AH uses the Level II ROM subroutine that both displays the ASCII byte in "A" register on the video display and automatically increments (adds 1) to the cursor location in MEM at 4020H. As such it is totally unnecessary for us to set up a separate data/file table to generate video position in either the TRANSMIT or RECEIVE video segments as long as we keep track of and move the cursor location as desired. A real time and memory saver. Now let's take few steps backwards.

TRSW = TRANSMIT & RECEIVE SWITCH:

195: In this line our CP = compare sets the Z flag if the subtraction of 31, which equals CLEAR key pressed, was zero.

200: Instructs the program to JUMP to TRSW (transmit-receive switch) IF the Z flag was set. The only difference between the JR and JP opcodes is the distance to jump and the fact that JR takes one less byte of MEM than JP. JR can only jump +129 or -126 memory locations, whereas JP can jump anywhere up to 65535 bytes in memory. Since every JR takes 1 byte less of memory to perform than a JP, using JR is a good habit to form as the excellent Radio Shack Editor/Assembler will tell you in no uncertain terms if you chose to JR 'OUT OF RANGE.'

410: Here we are at TRSW. You pressed the CLEAR key is why we are here. Lines 410 through 525 perform the simple function of a 'toggle switch.' If we were in TRANSMIT when the CLEAR key was pressed, these lines transfer us to RECEIVE and if we were in RECEIVE they transfer us to TRANSMIT. We must of necessity stash away the receive or transmit cursors' position and VIDEND of each segment so the program will know when to scroll if required. After doing all these good things in about 25 micro-seconds, line 530 returns us to the keyboard. It was a very short and fast trip. So quick that we will not even count it as a major or significant program segment.

205: In this line our compare sets the Z flag IF the down-arrow key was pressed, and line 210 sends the program off to line 535, RCLS, clear the screen of the receive segment. Line 535 is an error trap. We MUST be in the RECEIVE segment to CLS it. If not in RECEIVE, line 545 sends us back to the keyboard for further instructions. Assuming we know what we are doing and were in the RECEIVE sector when down-arrow was pressed, lines 550 through 570 utilize the Z-80 opcode LDIR to CLS the RECEIVE sector, only, after which lines 575 and 580 reset the RECEIVE cursor to the beginning of the 9th line. Line 585 simply sends us back to the keyboard for more instructions. Elapsed time = a few microseconds.

215: Is identical to the above, except for the TRANSMIT segment and the 'up-arrow' which calls the subroutine.

225: Compares the "A" register to 13 decimal which = the ENTER key pressed. If the subtraction was zero, line 230 zaps the program off to STORE the receive segment.

645: STORE is where we are now. Lines 645 through 655 check to see that we are indeed in RECEIVE mode, and send us back to the keyboard if we are NOT in RECEIVE. This is another operator error trap. If you choose to STORE both TRANSMIT & RECEIVE video segments, by all means take it out. Line 660 to 670 are our STORE counter. We chose to allow 5 RECEIVE video segments to be stored (so this demo program would work with 4K MEM TRS-80s). What happens if you inadvertently try to store a 6th RECEIVE video segment by pressing ENTER? Why at first, nothing at all. Line 675 sends the program off to OUT which resets the store COUNTer to zero and store MEM location to 32768 decimal (line 940 must also be changed when relocating the program), and then jumps back to the keyboard. No MEM has been stored and MOST importantly, number 5 STORE has not been lost.....yet. The clue is that NOTHING happened. The RECEIVE segment did not CLS and the STORE counter on line 1 did not change. This serves as a reminder to you that STORE MEM is full and IF you press ENTER again, the original #1 store will be wiped out and lost. It is as idiot proof as simplicity will allow. If we press ENTER a 2nd time, then former #1 STORE is discarded and the new RECEIVE segment automatically loaded into #1 STORE location and the program falls through line 675, OUT, to line 680.

680: Lines 680 through 735 are rather busy when called. They first display the STORE counter at the end of the first TRANSMIT line, second they move the RECEIVE video display segment to the proper storage PLACE, third they update the PLACE to store the next RECEIVE segment, and fourth & last the program jumps to receive CLS to let you know it has accomplished your wishes and is awaiting further instructions from the keyboard.

235: Compare (subtract) 9 from the "A" register is the keyboard subroutine's check to see if the right-arrow has been pressed which line 240 sends us off to RESTORe, if indeed it has.

740: RESTORe is a rather simple subroutine that may be shortened considerably, but since this is supposed to be an instructional program, let's leave it alone. Lines 740 and 745 check to make sure we are in the TRANSMIT mode to simplify cursor handling. Lines 755 & 760 display the '?' in the first position of the 2nd transmit line asking us, "what stored receive segment do we wish RESTORed?" Line 765 places the number we press on the keyboard in the "A" register, and lines 770 to 815 check to see which stored segment we wish RESTORed and then jump to ONE to FIVE in lines 855 to 925. These lines serve only to load SHOW memory

location (2 bytes at line 850) with the decimal location of the memory position called in line 765. Lines 855, 870, 885, 900, and 915 must also be changed when you relocate the program. Since we have already loaded SHOW memory location with the address we wish to RESTORe all that remains is to jump to TELL at line 825.

825: TELL again uses the Z-80 opcode LDIR, this time to move 511 bytes from SHOW to the RECEIVE segment of the video display, and then returns us to the keyboard.

TOTAL MODIFICATIONS REQUIRED 4K & 16K MEMORY SYSTEMS:

Rather than list them each and every one, just remember:

- For 16K MEM systems simply subtract 3000 decimal from those memory locations noted in the program review.

- For 4K MEM systems simply subtract 14871 decimal from those memory locations noted in the program review.

SUMMARY:

Split screen operation of the TRS-80 is in actuality much easier than most programmers realize. Though this subroutine appears rather lengthy it occupies only 584 bytes, so may be entered with the excellent Radio Shack Editor/Assembler in less than an hour's time.

How to use it in a real-time program? Again, quite simply. All you need do is change the input to the RECEIVE segment from the keyboard's "A" register to whatever output your receive program delivers, also in the "A" register. This Chapter is long enough for now. We'll have a detailed program, hopefully in a later Volume, which will illustrate its use in a real-time radio teletype and/or Morse code transmit & receive program with type-ahead interrupts et al.

Can this program be shortened? Of course it can. You learned an entirely different method of video MEM storage in the last Chapter which was MUCH shorter, and just as efficient. Have a go at using Chapter 7's MEM storage subroutine for this Chapter as it will save a considerable number of program bytes.

Why did we not use Chapter 7's MEM storage subroutine in this Chapter? A good question, Gridley. ANSWER: there are a zillion different ways to skin a cat (may my cats, Harlequin and R/C forgive me), and this Chapter's version may give YOU some ideas for other applications of this particular method.

Last thought: ok; we should be mixing mortar or carrying hods.

- CHAPTER 9 -

LEVEL II ROM CALL & ANCILLARY FUNCTION SUMMARY

ROM CALL ADDRESSES IN DECIMAL

BASIC FUNCTION	CALL ADDRESS	BASIC FUNCTION	CALL ADDRESS
ABS	2423	ASC	10767
ATN	5565	AUTO	8200
CDBL	2779	CHR$	10783
CINT	2687	CLEAR	7802
CLOAD	11295	CLS	457
CONT	7652	COS	5441
CSAVE	11253	CSNG	2737
DATA	7941	DEFDBL	7689
DEFINT	7683	DEFSNG	7686
DEFSTR	7680	DELETE	11206
DIM	9736	EDIT	11872
ELSE	7943	END	7598
ERL	9437	ERR	9423
ERROR	8180	EXP	5177
FIX	2854	FOR	7329
FRE	10196	GOSUB	7857
GOTO	7874	IF	8249
INKEY$	413	INP	10991
INPUT	8602	RET TO BASIC	114
INT	2871	LEFT$	10849
LEN	10755	LIST	11054
LOG	2057	LLIST	11049
LPRINT	8295	MEM	10185
MID$	10906	NEW	6985
NEXT	8886	NOT	9668
ON	8044	OUT	11003
PEEK	11434	POINT	307
POKE	11441	POS	10229
PRINT	8303	RANDOM	467
READ	8687	REM	7943
RESET	312	RESTORE	7569
RESUME	8111	RIGHT$	10897
RND	5321	RUN	7843
SET	309	SGN	2442
SIN	5447	SQR	5095
STOP	7593	STR$	10294
STRING$	10799	SYSTEM	690
TAN	5544	TROFF	7672
TRON	7671	USR	10238
VARPTR	9461	VAL	10949

KEYBOARD MEM LOCATIONS IN DECIMAL:

PEEK (14337) =	@	A	B	C	D	E	F	G
PEEK (14338) =	H	I	J	K	L	M	N	O
PEEK (14340) =	P	Q	R	S	T	U	V	W
PEEK (14344) =	X	Y	Z					
PEEK (14352) =	0	1	2	3	4	5	6	7
PEEK (14368) =	8	9	:	;	,	-	.	/
PEEK (14400) =	ENT	CLR	BRK	UA	DA	LA	RA	SPA
PEEK (14464) =	SHIFT							

VALUE =	1	2	4	8	16	32	64	128

The keyboard/switch matrix will output the VALUEs shown above when a single key is pressed at the corresponding MEM location. For multiple keys pressed simultaneously, add up the values for each key; i.e., "JKL" = 4 + 8 + 16 = 28 total at MEM location 14338 decimal.

- DATA MOVEMENT TABLE -

NO.	FROM	TO	CALL	NT(40AFH)
A.	ACCUM	STACK	09A4H/2468	2,4
B.	(HL)+	ACCUM	09B1H/2481	4
C.	BCDE	ACCUM	09B4H/2484	4
D.	ACCUM	BCDE	09BFH/2495	4
E.	(HL)+	BCDE	09C2H/2498	4
F.	ACCUM	(HL)+	09CBH/2507	4
G.	(DE)+	(HL)+	09CEH/2510	4
H.	(HL)+	(DE)+	09D2H/2514	2,4,8
I.	(DE)+	(HL)+	09D3H/2515	2,4,8
J.	(DE)+	(HL)+	09D6H/2518	A REG
K.	(DE)+	(HL)+	09D7H/2519	B REG
L.	"CS"	ACCUM	09F4H/2548	2,4,8
M.	ACCUM	"CS"	09FCH/2556	2,4,8
N.	HL	ACCUM	0A9AH/2714	2
O.	DE	HL	EX DE,HL	2
P.	HL	DE	EX DE,HL	2
Q.	BC	STACK	PUSH BC	2,4
R.	DE	STACK	PUSH DE	2,4
S.	HL	STACK	PUSH HL	2
T.	STACK	HL	POP HL	2
U.	STACK	DE	POP DE	2,4
V.	STACK	BC	POP BC	2,4

- COMPARE TABLE -

NO.	ITEM #1	SUBTRACT	ITEM #2	CALL	NT(40AFH)
A.	ACCUM	-	BCDE	0A0CH	4
B.	HL	-	DE	0A39H	2
C.	ACCUM	-	"CS"	0A4FH	8
D.	"CS"	-	ACCUM	0A78H	8
E.	ACCUM	DETERMINE SIGN		0994H	2,4,8
F.	HL	-	DE(unsigned)	1C90H	2

NOTE: No. E above is same as the BASIC SGN function, but returns to register "A": zero if ACCUM = 0, +1 if ACCUM greater than zero, and 255 (0FFH) if ACCUM is less than zero.

- DATA CONVERSIONS -

Are straightforward and very necessary in most all arithmetic operations as the NT (number type) must match-up with the CALL subroutine's function; i.e., integer, single precision or double precision + - * /. The most useful conversions are:

CALL 0A7FH: any ACCUM to integer ACCUM (CINT).
CALL 0AB1H: any ACCUM to single precision ACCUM (CSNG).
CALL 0ACCH: integer ACCUM to single precision ACCUM.
CALL 0ACFH: integer HL to single precision ACCUM.
CALL 0ADBH: any ACCUM to double precision ACCUM.
CALL 0E65H: ASCII string to ACCUM in double precision format.
CALL 0E6CH: ASCII string to ACCUM; NT will = minimum required.
CALL 0FBDH: ACCUM to ASCII string

- ARITHMETIC CALL SUMMARY -

	INTEGER NO.	SINGLE PRECISION	DOUBLE PRECISION
ADDITION	0BD2H/3026 DE+HL	0716H/1814 BCDE+ACCUM	0C77H/3191 ACCUM+"CS"
SUBTRACT	0BC7H/3015 DE-HL	0713H/1811 BCDE-ACCUM	0C70H/3184 ACCUM-"CS"
MULTIPLY	0BF2H/3058 DE*HL	0847H/2119 BCDE*ACCUM	08A2H/2210 ACCUM*"CS"
DIVIDE	2490H/10560 DE/HL	08A2H/2210 BCDE/ACCUM	0DE5H/3557 ACCUM/"CS"

NOTE: NT (number type) at (40AFH) must agree with operation CALLed. NT: 2 = integer, 3 = string, 4 = single precision, and 8 = double precision.

* ACCUM at MEM locations 411DH through 4124H.
* "CS" = CDBL Store at MEM 4127H through 412EH.

- CHAPTER 10 -

TRS-80 MORSE CODE TRANSMIT & RECEIVE PROGRAM BY W4UCH VER 2.2

Detailed operating instructions are presented in Part 1 of the program. The main program is Part 2 that also includes a 5 page instruction summary that may be called anytime from the TRANSMIT MODE by operators new to the system. The program documentation that follows will not duplicate in detail the instructions included in Part 1, but is provided for those users who "wish-to-dig-deeper" into this BASIC program's logic, program flow, and layout.

It is written in Level II BASIC which allows transmit and receive code speeds up to the 25 word per minute area, yet most importantly allows the average user to follow the program logic and flow and make his/her modifications to the program very easy to accomplish which machine code would not allow.

The most unique aspect of this program is that it will both transmit and receive Morse code on any standard Level II TRS-80 Microcomputer (or most any microcomputer that uses 8K and up Microsoft Basic with modest changes) WITHOUT any peripheral or ancillary devices whatsoever. The cassette motor control relay K1 is used for the keying relay and the cassette EAR plug line for receiving Morse code audio of approximately 1 volt peak to peak derived from the station receiver's speaker terminals.

A unique software solution renders the TRS-80 flip-flop Z-24 "invisible" to incoming signals. Since the TRS-80 cassette control relay K1 will only handle VERY low power levels (about 6 volts at 400-500 mils) it is STRONGLY recommended that a 7406/7507 TTL buffer chip or Radio Shack #275-004 ($2.99) relay be used as a buffer between the TRS-80 and the station transmitter.

For gaining program operating experience, a low cost Radio Shack Morse Code Practice Oscillator (#20-005) may be used for both generating the Morse code for the TRS-80 to decode and print out (using the earphone jack on the code oscillator connected to the TRS-80's EAR plug), and by installing a subminiature Radio Shack phone jack on the code oscillator across the keying circuit, and having the TRS-80 subminiature cassette REMOTE plug plugged into it for TRS-80 Morse output.

PROGRAM ORGANIZATION:
The program consists of eight segments:
1. Initialization (CLEAR, DIMENSION & DEFINE INTEGERS)
2. Transmit Morse look-up table (1=dot and 2=dash)
3. Transmit Morse timing
4. "Q" signal - prepared messages (20+)
5. Code practice (alphabet, alphanumerics, all + punctucation)
6. Receive Morse decoding algorithm
7. 100 page automatic logbook/file for contests, etc.
8. Instruction summary (5 pages for newcomers)

INITIALIZATION SEGMENT:

This segment allows the user to choose either alphanumeric readout or Morse code readout on the video display, to choose transmit Morse code speed (receive speed is automatic), and reminds the operator that "left-arrow" calls up the instruction summary and the "CLEAR" key is the transmit/receive switch. It also defines as integers A to Z for optimum code speed, DIMENSIONS the Morse receive array, and CLEARS 2550 bytes for the automatic logbook/file. Lastly, it includes an error trapping function that may be deliberately invoked to obtain immediate return to the TRANSMIT mode.

TRANSMIT MORSE LOOK-UP TABLE:

Surprisingly, the fastest means of generating Morse code using the 12K Microsoft BASIC in the Level II TRS-80 (no machine code allowed to facilitate user comprehension) is the simple IF-THEN statement and look-up table which converts the alphanumeric/punctuation symbol to a number in which all 1's = dots and all 2's = dashes, for its equivalent Morse code character; i.e., A=12, B=2111, C=2121, etc. Each character is followed by a GOTO directive which directs the program to the transmit Morse timing segment to save time. Though Version 2.2's transmit Morse look-up table is given alphabetically and numerically for convenience, it may be further speeded up by rearranging the alphanumerics in the same order as the DATA table in the RECEIVE Morse decoding segment, if desired.

In the RECEIVE mode, the order approximates the most commonly used letters in the English language, in order of usage, which was the way Morse code has evolved to today's standard; i.e, "E" is the most commonly used letter, so it = a dit (1), and "Q" the most infrequently used letter so it = dah dah di dah (2212). If one were to transpose the 1's to 0's and the 2's to 1's you would have binary numbers = to the most frequently used alphabet characters in English in proper order. Give you any ideas?

In the TRANSMIT MODE all generally accepted Morse characters are provided plus EOM (end of message) and EOW (end of work) by using the "#" and "&" symbols, respectively. Should a keyboard character such as "$" or "%" or "@" which has no Morse equivalent be be entered via the keyboard inadvertently, an error-trapping subroutine skips it and the program awaits the next legal alphanumeric with a Morse equivalent. At the end of the look-up table the ASCII codes for up-arrow, left-arrow, down-arrow, right-arrow, and CLEAR key are scanned and the programm directed to the subroutines of: "Q" signal-message, auto-logbook, instruction summary, log book review, or RECEIVE mode as called by YOU, the operator and Chief-Pilot of this exquisitely wonderful machine.

TRANSMIT MORSE TIMING SEGMENT:

Here is the segment that accomplishes the job of translating the Morse character's 1's and 2's (it could just as well have been 0's and 1's) into properly timed dots and dashes with the correct timing intervals between each element and character via the LEN and MID string functions which allows the program to 'peel off' each element of a character, one at a time. The international standard of: dot = 1 time interval, dash = 3 times 1 dot interval, and space between dot/dash elements of a given character = 1 dot interval, is set by this segment. Tranmitting code speed is determined by multiplying each element that forms a Morse character by "S" which is the adjusted value for the desired code speed that the operator INPUT during program initialization.

Upon completion of each Morse character, this segment then directs the program to the operator selected input; i.e., back to the keyboard, "Q" signal/message subroutine, RECEIVE mode, logbook subroutines, or instruction summary. Prior to output on video, this segment determines whether alphanumerics or Morse code was selected. Morse printout via 1's and 2's is seldom used except for the first few training periods with those individuals just beginning to learn Morse.

The cassette motor control relay K1, on the bottom left side of the lower TRS-80 printed circuit board (tubular yellow plastic enclosure), is closed and opened as the keying relay via the OUT (port) 255 statement. If the user is skilled in printed circuit board work, it is a simple matter to install a normally closed mini-phone jack on the rear of the TRS-80 keyboard in series between relay K1 and the output of integrated circuit Z-41 for transmit keying. Thus allowing Z-41 to drive a separate relay such as the Radio Shack #275-004 for transmitter keying. This relay will handle 125 volts ac at 1 amp and is fast enough to follow the program up to about 25 words per minute. Conversely, a high speed 5 to 6 volts dc reed relay may be used which will follow this program up to about 40 words per minute. Above this speed, program execution time in BASIC becomes the limiting factor. By utilizing the excellent Mumford Micro Systems 3 speed TRS-80 clock modification, both TRANSMIT and RECEIVE modes may be increased an additional 50 percent.

"Q" SIGNAL - PREPARED MESSAGE SEGMENT:

Twenty prepared "Q" signal and message formats are given including: CQ, QTH, QRZ, QRX, QSL, QSY, QSY+, QSY-, QRM, QRN, QRS, QRQ, RST, QSL, 73, etc. There is no limit to the number of additional messages that may be added, except available memory. There is also a SPEED subcommand which allows the operator to change transmit code speed without reinitializing the program and losing the data stored in the automatic logbook funtion. This segment also allows the operator to select the type of TRANSMIT Morse code practice desired.

Code 1 = alphabet only, Code 2 = alphanumerics, Code 3 = alphanumerics + punctuation. Though the arrow symbols are illustrated as reminders, they may only be used during the transmit or receive modes. This segment also uses the LEN and MID string functions of Level II BASIC for peeling off each letter, one at a time, for each prepared message. Each message is limited to a maximum of 240 bytes (string length), but by concatenating strings with appropiate software mods any message of any length may be transmitted. One final noteworthy subcommand included in this segment is the TEST subcommand. This function outputs the word PARIS with appropriate letter and word spacing standards so that the operator may time the number of words sent for 15 seconds, multiply by 4, and have his exact words per minute code speed calibration.

CODE PRACTICE SEGMENT:

This unique subroutine utilizes the random number generator incorporated in the Zilog Z-80 microprocessor to generate a number between 1 and 26 in the Code 1, alphabet only, code practice mode. By adding 59 to to the random number the ASCII character code for the alphabet from A to Z is generated and output, a letter at a time, in 5 letter code groups. Code 2, alphanumerics, is generated in much the same way by randomly generating a number from 1 to 47 and adding 48 to it to obtain the ASCII character code for both numbers and alpahbet. Since ASCII character codes 60, 61, 62, and 64 which equal less than, equal sign, greater than, and @, respectively, have no Morse code equivalents, they are trapped and not output.

Code 3, alphanumerics + punctuation is generated in much the same way. For brevity, the Morse double dash is displayed on video as a single dash, but for purists may be easily modified to a double dash if desired. Also, the normal 7 times dot length spacing after punctuation has been held to only 3 times dot length as it has been found in numerous Morse code training seesions that this convention speeds up the learning process. Spacing between each 5 letter group uses the international standard 7 times dot length for word spacing.

RECEIVE MORSE DECODING SEGMENT:

Utilizes an algorithm derived by the MIT Radio Club many years ago and improved upon by Robert Kurtz and the author. Its claim to fame is the method we developed to interface the TRS-80 with an ordinary communications receiever's speaker output that DOES NOT REQUIRE any ancillary/peripheral devices to work properly with Morse signals of S4 or stronger. Through sheer serendipity, cleverness, and lots of luck this subroutine makes the TRS-80 flip-flop Z-4 invisible to the approximately 1 volt peak to peak audio Morse signal coming from the station receiver's speaker terminals. This is done by re-setting flip-flop Z-4 every time the length of time is measured by the program to determine whether the Morse element is a dot, dash, or element space. With a good signal to noise ratio incoming signal (S4 or better), it will copy well sent

Morse up to 20 to 25 words per minute, which is about its upper limit due to BASIC (with standard clock) program execution time.

For operators working stations with "swing fists" (odd-ball dot/dash timing ratios), the RECEIVE MODE subroutine allows the operator to change these ratios by pressing "P" on the keyboard. This takes a bit of experimenting and experience, but after a few hours operating is quite easy to implement. For operating convenience, the FILE (auto-logbook) function may be called from both the transmit and receive modes.

There are many algorithms for decoing Morse code that may be written in Level II BASIC, but when the trade-offs between program length and execution time are evaluated, this version appears to be the best compromise. Z-4, an LM-3900 Norton operational amplifier in the TRS-80's cassette CASSIN input line is designed to serve as a pulse shaping and level adjusting network for the CLOAD function in normal TRS-80 operation. It works remarkably well for Morse code too, but may be improved a bit (for very weak signals) by adding an AVC/limiting amplifier between the TRS-80 and receiver speaker terminals as described by N6WA in the Sept. '79 issue of 73 magazine, pages 116-117, if an emitter follower 2N2222 transistor is added to match the TRS-80's 100 ohm input impedance.

FILE AND FILE-REVIEW SEGMENT:

This function is provided the operator to create a semiautomatic logbook with auto-sequencing for each entry. During initialization the program CLEARs 2550 bytes for this subroutine thus allowing only 25 bytes per entry if all 100 log entries were used. By all means CLEAR as many bytes as your installed memory will allow. This subroutine's most useful function is during CW (Morse code) amateur radio contests which is the reason for the auto-sequencing aspect of the program as time/speed are important. The FILE subroutine may called from both TRANSMIT and RECEIVE modes by pressing the 'right-arrow' on the keyboard. Each time it is called it will automatically advance to the next unused file where the call letters of the station worked, date, time, band, or what have you may be entered.

When the FILE REVIEW subroutine is called by pressing 'down-arrow' on the keyboard, the program will sequentially display 4 files per page (16 lines maximum if each of the 4 files is filled to capacity), each time the ENTER key is pressed. You do not have to review all 25 file pages (4 per page times 25 = 100 total) to return to the TRANSMIT mode, but may escape anytime by pressing 'break,' then '@,' and then ENTER. Here we deliberately induce an error and use the ON ERROR GOTO function to immediately put us back in the TRANSMIT MODE. This is a real time saver during CW contests.

At the end of a day's operation, or end of a contest, the file data may be saved on cassette or disk for permanent record and storage using the PRINT#-1, function described on page 3/10 (for cassette) of the Level II manual. If you plan to use this function frequently, by all means add the following lines to this program:
5000PRINT#-1,BA$:PRINT#-1,BB$:PRINT#-1,BC$:PRINT#-1,BD$ (etc).
Remember that each print statement will only handle strings that TOTAL 240 bytes. This is why the PRINT#-1, is repeated for each string we wish to CSAVE. Add as many lines and strings as you wish to CSAVE and then press 'BREAK,' enter RUN 5000 (with the cassette turned "ON" and RECORD and PLAY depressed) whenever you wish to make a permanent record of your logbook entries.

INSTRUCTION SUMMARY SEGMENT:
One usually does not write instructions on how to use instructions. The program's instruction summary is provided basically for the user new to the system who does not wish to pickup a written instruction during system operation. It is called from the TRANSMIT MODE by pressing 'left-arrow.'

HINTS AND KINKS FOR SUCCESSFUL SYSTEM OPERATION:
Probably the most difficult challenge presented to the user of this Morse Code System (or ANY TRS-80 Morse Code Program) will be the problem of quieting down the RFI (radio frequency interference) generated by the TRS-80 itself.

Every little digital gate in the TRS-80 plus the nominal 10.6445 MHz crystal oscillator and all the clock dividers are each and every one a miniature spark coil transmitter, or at the very least act like one. Do not let this bamboozle or overwhelm you. We shall overcome if we follow a few not too difficult ground rules that will allow us to easily copy most any Morse signal that we can hear. After July 1, 1980 all new microcomputers will have to meet the FCC rules regarding spurious radiation levels. Till then, try these recipes to minimize the problem:
1. Use Radio Shack #15-1106 line filters on EACH component's power line after cutting each power cord to minimum length.
2. Physically separate the TRS-80 at least 6 feet from the station receiver.
3. Run good quality well shielded (not the cheapest you can buy) RG8/U SEPARATELY from the transmitter and receiver to your antenna.
4. Your station antenna should be AT LEAST 80 feet away from the TRS-80. Install T/R relay and broadband preamplifier AT the ANTENNA. This is the most important item of all.
5. If all else fails, turn-off expansion interface when operating and DISCONNECT cassette and interface cables at the keyboard. When the operating day is finished, CSAVE your auto-logbook BEFORE powering up the interface and printing out the logbook data. NOTE: shielding & grounding all TRS-80 cables helps too. DO NOT BLAME THIS PROGRAM IF IT WILL COPY MORSE FROM YOUR CODE PRACTICE OSCILLATOR, BUT NOT FROM YOUR RECEIVER.....I.E., BETTER SHIELDING IS NECESSARY. 'Gud luk.'
Note: See March '80 QST pages 17-20, "Computer Interference."

W4UCH MORSE PROGRAM - PART 1

```
1 CLS
2 FORX=1TO16:PRINTCHR$(191);:NEXT:FORX=1TO3:PRINT" ";:NEXT:FORX=
1TO5:PRINTCHR$(191);:NEXT:FORX=1TO3:PRINT" ";:NEXT:FORX=1TO16:PR
INTCHR$(191);:NEXT:FORX=1TO3:PRINT" ";:NEXT:FORX=1TO5:PRINTCHR$(
191);:NEXT:PRINT:PRINT
3 FORX=1TO16:PRINTCHR$(191);:NEXT:PRINT"    ";:FORX=1TO16:PRINTCH
R$(191);:NEXT:PRINT"    ";:FORX=1TO5:PRINTCHR$(191);:NEXT:PRINT"
  ";:FORX=1TO16:PRINTCHR$(191);:NEXT:PRINT:IFA LT  3THENPRINT
4 A=A+1:IFA=4GOTO 5ELSEGOTO2
5 FORX=1TO5000:NEXT
6 CLS:PRINTCHR$(23):PRINT:PRINT:PRINT"   TRS-80 MORSE CODE SYSTE
M":PRINT:PRINT
7 PRINT"    OPERATING INSTRUCTIONS":PRINT:PRINT
8 PRINT"COPYRIGHT RICHCRAFT ENGINEERING":PRINT:PRINT:PRINT
9 PRINT"          BY W4UCH/2":PRINT:INPUTR:CLS
10 PRINT" GENERAL:"
11 PRINT"-"CHR$(34);"CONGRATULATIONS";CHR$(34);": YOU NOW HAVE T
HE FINEST TRS-80 MORSE SYSTEM   PROGRAM THAT REQUIRES NO ANCILLA
RY EXPENSIVE INTERFACES.":PRINT
12 PRINT"-IT OPERATES SIMPLY FROM THE CASSETTE INPUT AND OUTPUT
PORTS,    THUS REQUIRING ONLY A TRANSMITTER/RECEIVER OR CODE PRA
CTICE     KEYER/OSCILLATOR FOR OPERATION.":PRINT
13 PRINT"-FOR RECEIVE OPERATION ONLY A JACK IN PARALLEL WITH REC
EIVER     SPEAKER OUTPUT IS REQUIRED WHICH CONNECTS TO CASSETTE
EARPHONE  PLUG. JACK IS RADIO SHACK # 274-251 SET OF 3 FOR $ .99
 PER PAK."
14 PRINT"-FOR TRANSMIT MODE, THIS PROGRAM UTILIZES THE TRS-80 CA
SSETTE    MOTOR ON-OFF CONTROL RELAY AS A KEYING RELAY. RECOMMEN
DED MAX.  IS 6VDC AT 500 MA. OR BETTER YET USE R/S # 275-004 REL
AY. PRICE $ 2.99 EACH.    ";:INPUTR:CLS
15 PRINT" GENERAL CONTD."
16 PRINT"-THE TRS-80 CASSETTE MOTOR CONTROL PLUG MATES WITH SUBM
INIATURE  JACK, RADIO SHACK # 275-004 AT $ .99 FOR 2 EACH.  AS M
ENTIONED  ON THE 1ST PAGE, WE STRONGLY RECOMMEND YOU USE A BUFFE
R RELAY   OR TTL CHIP":PRINT
17 PRINT" TO PROTECT RELAY K1, WHOSE CONTACTS ARE UNRELIABLE AT
BEST     SEE INSTRUCTIONS FOR THIS SIMPLE AND INEXPENSIVE INSURANCE.
":PRINT
18 PRINT"-THERE ARE FOUR COMMONLY USED METHODS OF SENDING AND RE
CEIVING   MORSE CODE. THE FIRST, INVENTED BY SAMUEL MORSE OVER 1
00 YEARS  AGO USING A KEY & SOUNDER IS STILL IN USE TODAY BUT VI
A RADIO.":PRINT
19 PRINT"-THE SECOND METHOD USES DEDICATED TTL LOGIC I.C. CHIPS
TO BOTH   GENERATE AND DECODE MORSE.  THE ATRONICS # CR-101 CODE
 READER   AND CW KEYBOARD ARE TYPICAL AND PRICED AT $ 450. TOTAL
.";:INPUTR:CLS
20 PRINT"GENERAL CONTD:":PRINT
21 PRINT"-THE THIRD METHOD USES A DEDICATED MICROCOMPUTER TO GEN
ERATE AND DECODE BOTH MORSE AND TELETYPE. THE HAL COMMUNICATIONS
 CORP'S.  DS3000 KSR-3 IS TYPICAL AT $ 1575. PLUS $595. RTTY DEM
ODULATOR.";
22 PRINT" IT IS A SUPERB SYSTEM, AS WELL IT SHOULD BE, CONSIDERI
NG PRICE.  "
```

```
23 PRINT"-THE FOURTH METHOD UTILIZES A GENERAL PURPOSE MICROCOMP
UTER SUCH AS THE TRS-80 OR PET AND USES A COMBINATION MACHINE LA
NGUAGE    AND BASIC SOFTWARE PROGRAM THAT GENERATES AND DECODES
MORSE AS  WELL AS TELETYPE. ";
24 PRINT" IT REQUIRES A 9 I.C. PRINTED CIRCUIT BOARD  TO INTERF
ACE THE TRS-80 DATA BUS TO THE OUTSIDE REAL WORLD.":PRINT
25 PRINT"-AN OUTSTANDING EXAMPLE OF THIS APPROACH IS THE MACROTR
ONICS M-  80 INTERFACE SYSTEM DESIGNED BY DR. RON LODEWYCK N6EE,
 WHO ALSO WAS CO-AUTHOR OF THE TRS-80 LEVEL I & PET BASIC TUTORI
ALS.  ";:INPUTR:CLS
26 PRINT" GENERAL CONTD:":PRINT
27 PRINT"-THE MACROTRONICS M-80 IS AN OUTSTANDING EXAMPLE OF EXC
ELLENT    SOFTWARE AND ENGINEERING DESIGN. IT IS PRICED AT $129
POSTPAID. "
28 PRINT"-THE NEWEST APPROACH TO TRS-80 MORSE GENERATION AND DEC
ODING HAS BEEN DEVELOPED BY RICHCRAFT ENGINEERING AFTER EXTENSIV
E R & D   EFFORT.  THIS NEW SYSTEM REQUIRES NO INTERFACES WHATSO
EVER.     "
29 PRINT"-THESE INSTRUCTIONS WILL AUTO-LOAD AND AUTO-DELETE WHEN
 APPRO-   PRIATE IF YOU WILL KINDLY FOLLOW INSTRUCTIONS.":PRINT
30 PRINT:PRINT"      PRESS 'ENTER' TO CONTINUE   ";:INPUTR:CLS:P
RINT"GENERAL CONTD.":PRINT
32 PRINT"-THE TRS-80 MORSE CODE SYSTEM, BOTH TRANSMIT AND RECEIV
E MODES,  ARE ELEGANT IN THEIR SIMPLICITY.  MORSE CODE SPEEDS OF
 UP TO    30 - 35 WORDS PER MINUTE ARE EASILY ATTAINABLE WITH TH
IS PGM.    "
33 PRINT"-THE PROGRAM IS WRITTEN ENTIRELY IN TRS-80 LEVEL II MIC
ROSOFT    BASIC IN TWO VERSIONS.  VERSION 2.0 FOR 4K MEMORY TRS-
80S AND   VERSION 2.2 FOR 16K MEMORY TRS-80S.  BASIC DIFFERENCE
IS THE    AUTO-LOG, INSTR., & NO. OF Q-SIGNALS.":PRINT
34 PRINT"-THE ADVANTAGES OF A BASIC PROGRAM OVER ONE IN Z-80 MAC
HINE      LANGUAGE ARE MULTIFOLD; I.E., THE USER MAY EASILY MODI
FY THE    PROGRAM, AS DESIRED.  THE USER CAN EASILY UNDERSTAND T
HE PGM.   AND IMPROVE THEIR SKILLS.";:INPUTR:CLS
35 PRINT" THEORY OF THE SYSTEM:":PRINT
36 PRINT"-IN THE TRANSMIT MODE THE PROGRAM USES THE 'IF-THEN' FU
NCTION    WITH A 'LOOK-UP' TABLE TO FIND THE MORSE CHARACTER DES
IRED.     THE CHARACTER HAS 1 = DOT & 2 = DASH. FIRST THE CHARAC
TER IS    OUTPUT VIA MID$.":PRINT
37 PRINT"-TIMING IS ALWAYS 1 DASH=3 DOT LENGTHS; SPACE BETWEEN D
OTS AND   DASHES ALWAYS = 1 DOT LENGTH. SPACE BETWEEN WORDS ALWA
YS  = 7   DOT LENGTHS.  WORDS AND SENTENCES ARE SEQUENTIALLY OUT
PUT VIA   THE MID$ FUNCTION.":PRINT
38 PRINT"-CODE SPEED IS INTIALLY INPUT BY THE USER DURING INITIA
LIZATION  AND MAY BE CHANGED DURING TRANSMISSION BY THE 'SPEED'
COMMAND   THAT IS CALLED ALONG WITH 20 OTHER SUBCOMMANDS BY THE
[ CALL.  ";:INPUTR:CLS
39 PRINT" THEORY CONTD:":PRINT
40 PRINT"-IN THE RECEIVE MODE THE 1200-2000 CYCLE CW SIGNAL IS T
AKEN FROM THE RECEIVER'S 3-8 OHM SPEAKER IN PARALLEL. APPROXIMAT
ELY ONE   VOLT OF AUDIO A/C IS ADEQUATE. THIS IS INPUT TO THE TR
S-80 VIA  CASSETTE EAR PLUG.":PRINT
```

```
40 PRINT"-IN THE RECEIVE MODE THE 1200-2000 CYCLE CW SIGNAL IS T
AKEN FROM THE RECEIVER'S 3-8 OHM SPEAKER IN PARALLEL. APPROXIMAT
ELY ONE   VOLT OF AUDIO A/C IS ADEQUATE. THIS IS INPUT TO THE TR
S-80 VIA  CASSETTE EAR PLUG.":PRINT
41 PRINT"-THE CASSIN AUDIO INPUT IS FIRST PROCESSED BY Z4 QUAD O
P AMP AND OUTPUT TO Z24 SCHMITT TRIGGERS IN FLIP-FLOP CONFIGURAT
ION.      NORMALLY THIS WOULD BE THE 'END' OF THE GAME, BUT A UN
IQUE PGM. RENDERS Z24 INVISIBLE."P:PRINT
42 PRINT"-AFTER PASSING Z44 THE PROCESSED CW SIGNAL IS PLACED ON
 THE TRS- 80 D7 DATA LINE AS A DIGITAL 0 OR 1 WHERE ITS LENGTH I
S THEN    MEASURED BY THE SOFTWARE PROGRAM AND A SIMPLE ALGORITH
M DECODES THE SYMBOL.";:INPUTR:CLS
43 PRINT" THEORY CONTD:"
44 PRINT"-THE KEYBOARD 'CLEAR' KEY AUTOMATICALLY SWITCHES THE TR
S-80 FROM TRANSMIT TO RECEIVE AND VICE VERSA. THE 4 ARROW KEYBOA
RD KEYS   CALL SUBCOMMAND, FILE, FILE-REVIEW, AND INSTRUCTION SU
MMARY.     "
45 PRINT"-DURING TRANSMIT  MODE ALL FOUR ARROWS ARE OPERABLE. DU
RING THE  RECEIVE MODE ONLY ";CHR$(92);" AUTO-FILE AND 'CLEAR' M
AY BE USED.":PRINT
46 PRINT"-AUTO-FILE CONSISTS OF 100 AUTOMATICALLY SEQUENCED FILE
S, EACH   CAPABLE OF HOLDING UP TO 255 CHARACTERS. IT'S DESIGNED
 TO SERVE THE USER AS A LOGBOOK AND MAY BE RECORDED ON CASSETTE
ANY TIME.   "
47 PRINT"-FILE REVIEW, --";CHR$(94);" PRINTS OUT ALL 100 AUTO-FI
LE ENTRIES IN GROUPS OF 4 AT A TIME.":INPUTR:CLS
48 PRINT"          NOW PRESS 'ENTER' TO AUTO CLOAD
         MAIN PROGRAM.  THERE IS AN INSTRUCTION SUMMARY IN
         THE MAIN PROGRAM.   ";:INPUTR:CLS:CLOAD
```

- PART 2 -

```
49 REM FN IS RESERVED FOR DISC BASIC
50 CLS
51 PRINTCHR$(23):PRINT:PRINT"  TRS-80 MORSE CODE SYSTEM":PRINT
52 PRINT"    TRANSMIT AND RECEIVE":PRINT
53 PRINT"         VERSION 2.2":PRINT
54 PRINT"NO ANCILLARY DEVICES REQUIRED":PRINT
55 PRINT"         BY  W4UCH/2":PRINT:PRINT:PRINT:PRINT
56 PRINT"       COPYRIGHT "
57 PRINT"    ALL RIGHTS RESERVED   ";:INPUTR:CLS
58 CLEAR2550
59 DEFINTA-Z:CLS:INPUT"ALPHANUMERICS OR MORSE ON VIDEO DISPLAY (
A/M)";AA$:CLS:INPUT"CODE SPEED 5, 10, 15, 20, 25, 30, OR 35 W.P.
M.   ";S:CLS
60 IFS=35THENS=10ELSEIFS=30THENS=15ELSEIFS=25THENS=20ELSEIFS=20T
HENS=30ELSEIFS=15THENS=40ELSEIFS=10THENS=80ELSEIFS=5THENS=140ELS
EGOTO59
61 PRINT@195,"REMEMBER ";CHR$(125);"-- FOR INSTRUCTION SUMMARY I
N TRANSMIT MODE":INPUTR:CLS
62 PRINT@195,"REMEMBER THE 'CLEAR' KEY IS YOUR T/R SWITCH";:INPU
TR:CLS
```

```
63 DIMA$(100)
64 ON ERROR GOTO 65
65 RESUME 66
66 PRINT"TRANSMIT MODE"
67 Z=100:M$=""
68 A$=INKEY$:IFA$=""THEN68
69 IFA$=" "ANDLEN(D$) LT  1THENPRINT" ";:GOTO67
70 IFA$="A"THENM$="12":GOTO122
71 IFA$="B"THENM$="2111":GOTO122
72 IFA$="C"THENM$="2121":GOTO122
73 IFA$="D"THENM$="211":GOTO122
74 IFA$="E"THENM$="1":GOTO122
75 IFA$="F"THENM$="1121":GOTO122
76 IFA$="G"THENM$="221":GOTO122
77 IFA$="H"THENM$="1111":GOTO122
78 IFA$="I"THENM$="11":GOTO122
79 IFA$="J"THENM$="1222":GOTO122
80 IFA$="K"THENM$="212":GOTO122
81 IFA$="L"THENM$="1211":GOTO122
82 IFA$="M"THENM$="22":GOTO122
83 IFA$="N"THENM$="21":GOTO122
84 IFA$="O"THENM$="222":GOTO122
85 IFA$="P"THENM$="1221":GOTO122
86 IFA$="Q"THENM$="2212":GOTO122
87 IFA$="R"THENM$="121":GOTO122
88 IFA$="S"THENM$="111":GOTO122
89 IFA$="T"THENM$="2":GOTO122
90 IFA$="U"THENM$="112":GOTO122
91 IFA$="W"THENM$="122":GOTO122
92 IFA$="V"THENM$="1112":GOTO122
93 IFA$="X"THENM$="2112":GOTO122
94 IFA$="Y"THENM$="2122":GOTO122
95 IFA$="Z"THENM$="2211":GOTO122
96 IFA$="/"THENM$="21121":GOTO122
97 IFA$="."THENM$="121212":GOTO122
98 IFA$="?"THENM$="112211":GOTO122
99 IFA$=","THENM$="221122":GOTO122
100 IFA$="%"THENM$="11111111":GOTO122
101 IFA$="-"THENM$="21112":GOTO122
102 IFA$=":"THENM$="222111":GOTO122
103 IFA$=";"THENM$="212121":GOTO122
104 IFA$="#"THENM$="12121":GOTO122:REM "EOM"
105 IFA$="&"THENM$="111212":GOTO122:REM "Eo WORK"
106 IFA$="↑"THENGOTO126
107 IFA$="0"THENM$="22222":GOTO122
108 IFA$="1"THENM$="12222":GOTO122
109 IFA$="2"THENM$="11222":GOTO122
110 IFA$="3"THENM$="11122":GOTO122
111 IFA$="4"THENM$="11112":GOTO122
112 IFA$="5"THENM$="11111":GOTO122
113 IFA$="6"THENM$="21111":GOTO122
114 IFA$="7"THENM$="22111":GOTO122
115 IFA$="8"THENM$="22211":GOTO122
```

```
116 IFA$="9"THENM$="22221":GOTO122
117 IFA$=CHR$(8)GOTO317
118 IFA$=CHR$(31)GOTO171
119 IFA$=CHR$(9)GOTO310
120 IFA$=CHR$(10)GOTO209
121 IFA$=""ORM$=""GOTO67
122 IFAA$="A"THENZ$=A$ELSEZ$=M$+" "
123 FORK=1TOLEN(M$):C$=MID$(M$,K,1):IFC$="2"THENC=S*3ELSEC=S
124 OUT255,4:FORA=1TOC:NEXT:OUT255,0:FORA=1TOS:NEXT:NEXTK:PRINTZ
$;:FORA=1TOC:NEXT
125 IFZ=0THENNEXTLANDGOTO159ELSEIFZ=100GOTO67ELSEIFZ=1313GOTO165
ELSEIFZ=1306GOTO163ELSEIFZ=1310GOTO164
126 CLS:PRINT "  Q SIGNAL - MESSAGE - SUBCOMMANDS ":PRINT
127 PRINT"- CQ","- QTH","- QRZ","- QSL":PRINT
128 PRINT"- RST","- 73","- QRM","- QRN":PRINT
129 PRINT"- QRS","- QRQ","- QRX","- QRT":PRINT
130 PRINT"- QSB","- QSY","- QSY+","- QSY-":PRINT:PRINT"- CQSS","
- QRZSS","- SECT","- TEST":PRINT
131 PRINT"- CODE1 ","- CODE2 ","- CODE3 ","- SPEED":PRINT
132 PRINTCHR$(92);" FILE","-";CHR$(94); "REVIEW FILE",CHR$(91);"
 SUBCOMMAND",CHR$(93);"-";"INSTRUCTIONS"
133 PRINT@40,"";:INPUT"SIGNAL/MESSAGE";X$:Z=0:E=-1:CLS
134 IFX$="CQ"THEND$="CQ CQ CQ DE W4UCH/2 K "
135 IFX$="QTH"THEND$="QTH IS BOX 1065, CHAUTAUQUA LAKE, N.Y.   1
4722 "
136 IFX$="QRZ"THEND$="QRZ QRZ QRZ DE W4UCH/2 K "
137 IFX$="QSL"THEND$="QSL QSL GD LUK IN SS. DE W4UCH/2 K "
138 IFX$="RST"THEND$="RST 599 599 IN WNY WNY DE W4UCH/2 K "
139 IFX$="73"THEND$="73 TO U AND URS. HP TO WK U AGN SOON. DE W4
UCH/2 K "
140 IFX$="QRM"THEND$="QRM QRM PSE TRY AGN. DE W4UCH/2 K "
141 IFX$="QRN"THEND$="LOCAL QRN QRN PSE TRY AGN DE W4UCH/2 K "
142 IFX$="QRS"THEND$="QRS. PSE SLOW DOWN A BIT. DE W4UCH/2 K "
143 IFX$="QRQ"THEND$="QRQ. MAY I SPEED UP? DE W4UCH/2 K "
144 IFX$="QRX"THEND$="QRX. THE PHONE PSE STANDBY. "
145 IFX$="QRT"THEND$="MUST QRT NOW. 73, DE W4UCH/2 K "
146 IFX$="QSB"THEND$="QSB TERRIBLE PSE TRY AGN. DE W4UCH/2 K "
147 IFX$="QSY"THEND$="QRM WHERE U WISH TO MOVE? DE W4UCH/2 K "
148 IFX$="QSY+"THEND$="QSY UP 3KHZ UP 3KHZ NOW. DE W4UCH/2 K "
149 IFX$="QSY-"THEND$="QSY DOWN 3KHZ 3KHZ NOW. DE W4UCH/2 K "
150 IFX$="CQSS"THEND$="CQSS CQSS CQSS DE W4UCH/2 K "
151 IFX$="QRZSS"THEND$="QRZ SS QRZ SS QRZ SS DE W4UCH/2 K "
152 IFX$="SECT"THEND$="WNY WNY WNY DE W4UCH/2 K "
153 IFX$="TEST"THEND$="PARIS PARIS PARIS PARIS PARIS PARIS PARIS
 PARIS PARIS PARIS PARIS "
154 IFX$="CODE1"THENZ=1313:GOTO165
155 IFX$="CODE2"THENZ=1310:GOTO164
156 IFX$="CODE3"THENZ=1306:GOTO163
157 IFX$="SPEED"THENS=0:INPUT"NEW CODE SPEED";S:GOTO60
158 IFX$="↑"THEND$="AA":GOTO126
159 FORL=1TOLEN(D$):A$=MID$(D$,L,1):IFL=LEN(D$)THEND$="":GOTO67
160 IFA$=" "THENPRINT" ";:FORA=1TO7*S:NEXT:GOTO125
161 GOTO70
162 REM"CODE PRACTICE"
163 D=RND(47)+43:IFD=60ORD=61ORD=62ORD=64GOTO163ELSEGOTO166
164 D=RND(42)+47:IFD=58ORD=59ORD=60ORD=61ORD=62ORD=63ORD=64GOTO1
64ELSEGOTO166
```

```
165 D=RND(26)+64
166 A$=CHR$(D):E=E+1:IFE GT  4GOTO167ELSEGOTO70
167 FORF=1TO7*S:NEXT:PRINT" ";:E=0:GOTO70
168 REM "W4UCH/2 SPECIAL TRS-80 MORSE DECODER PGM LINES 2000 - 2
150"
169 INPUT"CHARACTER TIMING: 5 NOMINAL";CC
170 INPUT"SPACE TIMING: 3 NOMINAL";SS:GOTO173
171 PRINT"RECEIVE MODE":RESTORE:CC=5:SS=3
172 FORN=1TO100:READA$(N):NEXTN
173 A=INP(255)
174 C$=INKEY$:IFC$="P"THENGOTO169
175 IFC$=CHR$(31)GOTO66
176 IFC$=CHR$(10)GOTO209
177 IFA=127THEN173
178 B=0
179 IFA=255THENOUT255,0
180 A=INP(255):B=B+10
181 IFA=127THENC=((5*C)+(2*B))/6:DO=2*DO:DA=2*DA:DO=DO+1:GOTO188

182 IFB LT  (.5*C)THEN179
183 DO=2*DO:DA=2*DA:DA=DA+1
184 IFA=255THENOUT255,0
185 A=INP(255):B=B+10
186 IFA=255THENGOTO184
187 C=((4*C)+B)/5
188 B=0
189 IFA=255THENOUT255,0
190 A=INP(255):B=B+CC
191 IFA=255THENGOTO178
192 IFB LT  (.5*C)THENGOTO189
193 GOSUB199
194 A=INP(255):B=B+SS
195 IFA=255THENGOTO178
196 IFB LT  (2*C)THENGOTO194
197 PRINT" ";
198 GOTO173
199 DA=DA*2
200 D=DA+DO
201 IFD GT  100THEND=100
202 PRINTA$(D);
203 DA=0:DO=0
204 RETURN
205 DATAE,T,I,A,N,M,S,U,R,W,D,K,G,O,H,V,F,-,L,-,P,J,B,X,C
206 DATAY,Z,Q,-,-,5,4,-,3,-,-,-,2,-,-,-,-,-,-,-,1,6,-,/,-
207 DATA-,-,-,-,7,-,-,-,8,-,9,0,-,-,-,-,-,-,-,-,-,-,-,-,?
208 DATA-,-,-,-,-,-,-,-,.,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-
209 PRINT"AUTOMATIC FILE SUBROUTINE - ENTER WILL RETURN TO TRANS
MIT MODE. EACH FILE MAY CONTAIN UP TO 255 CHARACTERS.":PRINT
210 IFBA$ LT  "1"INPUTBA$:CLS:GOTO66
211 IFBB$ LT  "1"INPUTBB$:CLS:GOTO66
212 IFBC$ LT  "1"INPUTBC$:CLS:GOTO66
213 IFBD$ LT  "1"INPUTBD$:CLS:GOTO66
214 IFBE$ LT  "1"INPUTBE$:CLS:GOTO66
215 IFBF$ LT  "1"INPUTBF$:CLS:GOTO66
216 IFBG$ LT  "1"INPUTBG$:CLS:GOTO66
217 IFBH$ LT  "1"INPUTBH$:CLS:GOTO66
```

```
218 IFBI$ LT  "1"INPUTBI$:CLS:GOTO66
219 IFBJ$ LT  "1"INPUTBJ$:CLS:GOTO66
220 IFBK$ LT  "1"INPUTBK$:CLS:GOTO66
221 IFBL$ LT  "1"INPUTBL$:CLS:GOTO66
222 IFBM$ LT  "1"INPUTBM$:CLS:GOTO66
223 IFBN$ LT  "1"INPUTBN$:CLS:GOTO66
224 IFBO$ LT  "1"INPUTBO$:CLS:GOTO66
225 IFBP$ LT  "1"INPUTBP$:CLS:GOTO66
226 IFBQ$ LT  "1"INPUTBQ$:CLS:GOTO66
227 IFBR$ LT  "1"INPUTBR$:CLS:GOTO66
228 IFBS$ LT  "1"INPUTBS$:CLS:GOTO66
229 IFBT$ LT  "1"INPUTBT$:CLS:GOTO66
230 IFBU$ LT  "1"INPUTBU$:CLS:GOTO66
231 IFBV$ LT  "1"INPUTBV$:CLS:GOTO66
232 IFBW$ LT  "1"INPUTBW$:CLS:GOTO66
233 IFBX$ LT  "1"INPUTBX$:CLS:GOTO66
234 IFBY$ LT  "1"INPUTBY$:CLS:GOTO66
235 IFBZ$ LT  "1"INPUTBZ$:CLS:GOTO66
236 IFCA$ LT  "1"INPUTCA$:CLS:GOTO66
237 IFCB$ LT  "1"INPUTCB$:CLS:GOTO66
238 IFCC$ LT  "1"INPUTCC$:CLS:GOTO66
239 IFCD$ LT  "1"INPUTCD$:CLS:GOTO66
240 IFCE$ LT  "1"INPUTCE$:CLS:GOTO66
241 IFCF$ LT  "1"INPUTCF$:CLS:GOTO66
242 IFCG$ LT  "1"INPUTCG$:CLS:GOTO66
243 IFCH$ LT  "1"INPUTCH$:CLS:GOTO66
244 IFCI$ LT  "1"INPUTCI$:CLS:GOTO66
245 IFCJ$ LT  "1"INPUTCJ$:CLS:GOTO66
246 IFCK$ LT  "1"INPUTCK$:CLS:GOTO66
247 IFCL$ LT  "1"INPUTCL$:CLS:GOTO66
248 IFCM$ LT  "1"INPUTCM$:CLS:GOTO66
249 IFCN$ LT  "1"INPUTCN$:CLS:GOTO66
250 IFCO$ LT  "1"INPUTCO$:CLS:GOTO66
251 IFCP$ LT  "1"INPUTCP$:CLS:GOTO66
252 IFCQ$ LT  "1"INPUTCQ$:CLS:GOTO66
253 IFCR$ LT  "1"INPUTCR$:CLS:GOTO66
254 IFCS$ LT  "1"INPUTCS$:CLS:GOTO66
255 IFCT$ LT  "1"INPUTCT$:CLS:GOTO66
256 IFCU$ LT  "1"INPUTCU$:CLS:GOTO66
257 IFCV$ LT  "1"INPUTCV$:CLS:GOTO66
258 IFCW$ LT  "1"INPUTCW$:CLS:GOTO66
259 IFCX$ LT  "1"INPUTCX$:CLS:GOTO66
260 IFCY$ LT  "1"INPUTCY$:CLS:GOTO66
261 IFCZ$ LT  "1"INPUTCZ$:CLS:GOTO66
262 IFDA$ LT  "1"INPUTDA$:CLS:GOTO66
263 IFDB$ LT  "1"INPUTDB$:CLS:GOTO66
264 IFDC$ LT  "1"INPUTDC$:CLS:GOTO66
265 IFDD$ LT  "1"INPUTDD$:CLS:GOTO66
266 IFDE$ LT  "1"INPUTDE$:CLS:GOTO66
267 IFDF$ LT  "1"INPUTDF$:CLS:GOTO66
268 IFDG$ LT  "1"INPUTDG$:CLS:GOTO66
269 IFDH$ LT  "1"INPUTDH$:CLS:GOTO66
270 IFDI$ LT  "1"INPUTDI$:CLS:GOTO66
271 IFDJ$ LT  "1"INPUTDJ$:CLS:GOTO66
```

```
272 IFDK$ LT  "1"INPUTDK$:CLS:GOTO66
273 IFDL$ LT  "1"INPUTDL$:CLS:GOTO66
274 IFDM$ LT  "1"INPUTDM$:CLS:GOTO66
275 IFDN$ LT  "1"INPUTDN$:CLS:GOTO66
276 IFDO$ LT  "1"INPUTDO$:CLS:GOTO66
277 IFDP$ LT  "1"INPUTDP$:CLS:GOTO66
278 IFDQ$ LT  "1"INPUTDQ$:CLS:GOTO66
279 IFDR$ LT  "1"INPUTDR$:CLS:GOTO66
280 IFDS$ LT  "1"INPUTDS$:CLS:GOTO66
281 IFDT$ LT  "1"INPUTDT$:CLS:GOTO66
282 IFDU$ LT  "1"INPUTDU$:CLS:GOTO66
283 IFDV$ LT  "1"INPUTDV$:CLS:GOTO66
284 IFDW$ LT  "1"INPUTDW$:CLS:GOTO66
28' IFDX$ LT  "1"INPUTDX$:CLS:GOTO66
286 IFDY$ LT  "1"INPUTDY$:CLS:GOTO66
287 IFDZ$ LT  "1"INPUTDZ$:CLS:GOTO66
288 IFEA$ LT  "1"INPUTEA$:CLS:GOTO66
289 IFEB$ LT  "1"INPUTEB$:CLS:GOTO66
290 IFEW$ LT  "1"INPUTEW$:CLS:GOTO66
291 IFEX$ LT  "1"INPUTEX$:CLS:GOTO66
292 IFEY$ LT  "1"INPUTEY$:CLS:GOTO66
293 IFEZ$ LT  "1"INPUTEZ$:CLS:GOTO66
294 IFFA$ LT  "1"INPUTFA$:CLS:GOTO66
295 IFFB$ LT  "1"INPUTFB$:CLS:GOTO66
296 IFFC$ LT  "1"INPUTFC$:CLS:GOTO66
297 IFFD$ LT  "1"INPUTFD$:CLS:GOTO66
298 IFFE$ LT  "1"INPUTFE$:CLS:GOTO66
299 IFFF$ LT  "1"INPUTFF$:CLS:GOTO66
300 IFFG$ LT  "1"INPUTFG$:CLS:GOTO66
301 IFFH$ LT  "1"INPUTFH$:CLS:GOTO66
302 IFFI$ LT  "1"INPUTFI$:CLS:GOTO66
303 IFFJ$ LT  "1"INPUTFJ$:CLS:GOTO66
304 IFFK$ LT  "1"INPUTFK$:CLS:GOTO66
305 IFFL$ LT  "1"INPUTFL$:CLS:GOTO66
306 IFFM$ LT  "1"INPUTFM$:CLS:GOTO66
307 IFFO$ LT  "1"INPUTFO$:CLS:GOTO66
308 IFFP$ LT  "1"INPUTFP$:CLS:GOTO66
309 IFFQ$ LT  "1"INPUTFQ$:CLS:GOTO66
310 PRINT"FILE REVIEW - FOUR FILES PER PAGE:":INPUTR:CLS
311 PRINTBA$:PRINTBB$:PRINTBC$:PRINTBD$:INPUTR:CLS:PRINTBE$:PRIN
TBF$:PRINTBG$:PRINTBH$:INPUTR:CLS:PRINTBI$:PRINTBJ$:PRINTBK$:PRI
NTBL$:INPUTR:CLS:PRINTBM$:PRINTBN$:PRINTBO$:PRINTBP$:INPUTR:CLS:
PRINTBQ$:PRINTBR$:PRINTBS$:PRINTBT$:INPUTR:CLS
312 PRINTBU$:PRINTBV$:PRINTBW$:PRINTBX$:INPUTR:CLS:PRINTBY$:PRIN
T BZ$:PRINTCA$:PRINTCB$:INPUTR:CLS:PRINTCC$:PRINTCD$:PRINTCE$:PR
INTCF$:INPUTR:CLS:PRINTCG$:PRINTCH$:PRINTCI$:PRINTCJ$:INPUTR:CLS
:PRINTCK$:PRINTCL$:PRINTCM$:PRINTCN$:INPUTR:CLS
313 PRINTCO$:PRINTCP$:PRINTCQ$:PRINTCR$:INPUTR:CLS:PRINTCS$:PRIN
TCT$:PRINTCU$:PRINTCV$:INPUTR:CLS:PRINTCW$:PRINTCX$:PRINTCY$:PRI
NTCZ$:INPUTR:CLS:PRINTDA$:PRINTDB$:PRINTDC$:PRINTDD$:INPUTR:CLS:
PRINTDE$:PRINTDF$:PRINTDG$:PRINTDH$:INPUTR:CLS
314 PRINTDI$:PRINTDJ$:PRINTDK$:PRINTDL$:INPUTR:CLS:PRINTDM$:PRIN
TDN$:PRINTDO$:PRINTDP$:INPUTR:CLS:PRINTDQ$:PRINTDR$:PRINTDS$:PRI
NTDT$:INPUTR:CLS:PRINTDU$:PRINTDV$:PRINTDW$:PRINTDX$:INPUTR:CLS:
PRINTDY$:PRINTDZ$:PRINTEA$:PRINTEB$:INPUTR:CLS
```

```
315 PRINTEW$:PRINTEX$:PRINTEY$:PRINTEZ$:INPUTR:CLS:PRINTFA$:PRIN
TFB$:PRINTFC$:PRINTFD$:INPUTR:CLS:PRINTFE$:PRINTFF$:PRINTFG$:PRI
NTFH$:INPUTR:CLS:PRINTFI$:PRINTFJ$:PRINTFK$:PRINTFL$:INPUTR:CLS:
PRINTFM$:PRINTFO$:PRINTFP$:PRINTFQ$:INPUTR:CLS
316 PRINT"END OF FILE - ENTER TO RETURN TO TRANSMIT MODE":INPUTR
:CLS:GOTO66
317 CLS:PRINT"     I N S T R U C T I O N   S U M M A R Y  ";CHR$
(125);"--":PRINT
318 PRINT"-THE ↑ SYMBOL CALLS SUBCOMMAND ROUTINES DURING TRANSMI
T MODE.":PRINT
319 PRINT"-THE ";CHR$(92);" SYMBOL CALLS INPUT/FILE FUNCTION THA
T IS YOUR LOGBOOK.":PRINT
320 PRINT"-INPUT/FILE MAY BE CALLED FROM BOTH TRANSMIT AND RECEI
VE MODES.":PRINT
321 PRINT"-THE -";CHR$(94);" SYMBOL CALLS THE REVIEW FILE FUNCTI
ON.":PRINT
322 PRINT"-CODE1=ALPHABET : CODE2=ALPHANUMERIC : CODE3=CODE2+PUN
CTUATION.":PRINT
323 PRINT"-SPEED SUBCOMMAND CHANGES XMIT MORSE CODE SPEED AS DES
IRED.":PRINT
324 PRINT"-TEST SUBCOMMAND UTILIZES INTL. STD. CODE SPEED MEASUR
EMENT.":PRINT@987," ";:INPUTR:CLS
325 PRINT" INSTRUCTION SUMMARY-CONTINUED:":PRINT
326 PRINT"-RECEIVE MODE SOFTWARE ADJUSTS SPEED AFTER 6 TO 10 CHA
RACTERS.":PRINT
327 PRINT"-YOU MAY MODIFY RECEIVE CHARACTER & WORD SPACING BY PR
ESSING P.":PRINT
328 PRINT"-TRS 80 WILL COPY ITS OWN TAPED MORSE NEAR PERFECTLY W
ITH LINES  2005 - CC=6 & SS=6 AND LINE 2045 B LT  (.3*C) CHANGED
 ACCORDINGLY.  "
329 PRINT"-RELAY K1 ON LEFT EDGE OF Z-80 PCB IS USED FOR XMIT KE
YING.":PRINT
330 PRINT"-IT IS NORMALLY THE CASSETTE MOTOR CONTROL RELAY 6VDC
500 MA.":PRINT
331 PRINT"-K1 IS UNFORTUNATELY UNRELIABLE WITH A POOR MTBF USED
NORMALLY.":PRINT
332 PRINT"-FOR RELIABILITY CHANGE K1 TO RADIO SHACK # 275-004 $2
.99 EA.";:INPUTR:CLS
333 PRINT" INSTRUCTION SUMMARY - CONTINUED":PRINT
334 PRINT"-AUTOMATIC CASSETTE 'RECORD' AND 'PLAYBACK' OF INPUT-F
ILE DATA   OPTIONAL SOFTWARE PROGRAM IS IN INSTRUCTION MANUAL. C
ONCEPT IS  TO CSAVE INPUT-FILE / LOGBOOK ENTRIES AT END OF OPERA
TING DAY.  "
335 PRINT"-EQUIVALENT 'SPEED ' TABLE - WORDS PER MINUTE: (AUTOMA
TIC 16K    MEM VERSION AND MANUAL, SLIGHTLY SLOWER, FOR 4K MEM V
ERSION."
336 PRINT" INPUT","WPM","INPUT","WPM"
337 PRINT"  10","35"," 80","10"
338 PRINT"  20","25"," 90"," 9"
339 PRINT"  30","20","100"," 8"
340 PRINT"  40","15","110"," 7"
341 PRINT"  50","14","120"," 6"
342 PRINT"  60","12","140"," 5"
343 PRINT"  70","11","150"," 4     ";:INPUTR:CLS
```

```
344 PRINT" INSTRUCTION SUMMARY - CONTD:"
345 PRINT"-THOUGH ARROW COMMANDS ARE LISTED IN SUBCOMMAND TABLE,
 THEY MAY  ONLY BE CALLED IN TRANSMIT AND/OR RECEIVE  OPERATING
MODES.     "
346 PRINT"-ANY NEW SYSTEM REQUIRES PRACTICE AND EXPERIENCE TO UT
ILIZE ITS  CAPABILITES FULLY.  IT IS SUGGESTED THAT YOU GAIN EXP
ERIENCE    WITH THE TRS-80 MORSE CODE SYSTEM USING THE RADIO SHA
CK MODEL  ";
347 PRINT" # 20-005 CODE PRACTICE OSCILLATOR FOR TRS-80 INPUT AN
D OUTPUT,  USING YOUR OWN KEY TO SEND.  ADJUST VOLUME FOR ACCURA
TE COPY.   "
348 PRINT"-IN THE FUTURE, A CASSETTE LOGBOOK MAY BECOME A WORLD
STANDARD.  FOR TRS-80 OWNERS AND OPERATORS, THE 'FUTURE ' IS NOW
       "
349 PRINT"-LINE 50 'CLEARS 2550 ' BYTES FOR THE 100 EACH AUTO-FI
LES.  IF   YOU HAVE ADEQUATE MEMORY, INCREASE THIS UP TO 25500 T
O FULLY    UTILIZE EACH FILE'S CAPABILITY OF 255 BYTES CAPACITY.
     ";:INPUTR:CLS
350 PRINT"-TO ESCAPE FROM: SUBCOMMAND, FILE, FILE-REVIEW, OR INS
TRUCTION   SUMMARY, PRESS 'BREAK', THEN '@', AND THEN 'ENTER' FO
R NEAR     IMMEDIATE RETURN TO THE 'TRANSMIT-MODE'":PRINT
351 PRINT"-PROGRAM SUGGESTIONS AND IMPROVEMENTS ARE ALWAYS WELCO
ME.":PRINT
352 PRINT"-PLEASE SEND TO:":PRINT
353 PRINT"                  RICHCRAFT  ENGINEERING
354 PRINT"                  DRAWER 1065"
355 PRINT"                  CHAUTAUQUA, N.Y. 14722
356 PRINT:PRINT:PRINT"          'ENTER ' WILL RETURN TO TRANSMIT
 MODE    ";:INPUTR:CLS:GOTO66
357 :
358 END- - - NOTE - - - LT = < and GT = > - - - NOTE - -
```

ASIDE TO THE AUDIENCE:

This program was written circa 1977/1978 "BD." BD = 'before disks' were available for the TRS-80. Can you improve upon it?

Sure you can. Try rearranging lines 70 - 116 in the same relative order as lines 205-208. Average XMIT speed increase?

There are at 'least' 2 other easily written algorithms for decoding Morse that may be written in BASIC including: IF - THEN and a lookup table arranged in numerical order = 1, 2, 11, 21, 22, 111, 112, 122, etc. If you find/invent other algorithms for decoding Morse in BASIC, please let us know as it is our hobby within a hobby within a hobby. We would also appreciate hearing about any SIMPLE Morse decoding assembly language program you may have written that is loaded into MEM using BASIC READ - DATA statements. Your REWARD? Beating us to the punch.

- CHAPTER 11 -

TV SATELLITE - AZIMUTH - ELEVATION - RANGE - TRS-80 PROGRAM

Here is an interesting program in BASIC for the TRS-80 micro that will give you the azimuth, elevation, and range to those big TV relay satellites in geostationary orbit from your home. This article is only a minor supplement to Bob Cooper, W5KHT's excellent TV Satellite article in the November '79 issue of 73 MAGAZINE, pp 120-133, if you wish to dig deeper.

Just like 432 MHz DX'ing with narrow beam antennas, "you can't work them if you can't find them." The problem at 4 GHz is considerably greater than 432 MHz since we are dealing with 8', 10', and 12' diameter parabolic reflector antennas with beamwidths less than 1 degree. This simple software program will also run on the Pet, Apple, KIM, Heathkit, etc. micros with only minor modification as they all use the excellent BASIC written by Microsoft.

The TV satellites we are primarily interested in from the western hemisphere are:

SATCOM 1	135 DEGREES W	-	SATCOM 2	119 DEGREES W
SATCOM 3 *	132 DEGREES W	-	COMSTAR 1	128 DEGREES W
COMSTAR 2	95 DEGREES W	-	COMSTAR 3	87 DEGREES W
WESTAR 1	87 DEGREES W	-	WESTAR 2	123.5 DEGREES W
WESTAR 3	91 DEGREES W	-	ANIK 1	91 DEGREES W
ANIK B	109 DEGREES W	-	ANIK 3	114 DEGREES W
STATSIONAR 4	14 DEGREES W	-	STATSIONAR 8	25 DEGREES W
STATSIONAR 10	170 DEGREES W			

All the above birds are operational except for SATCOM 3 which was launched early December '79 and was either "lost" or blew up when being injected into equatorial orbit, and STATSIONARs 8 and 10 which are due for 1980 launch.

THE PROGRAM:
This program is designed to output both to the video display and line printer. All that is required of the user is to input his/her location's latitude and longitude in degrees and minutes and then the program does the rest. Even though lines 150 and 170 printout "NORTH" and "WEST" they are just window dressing to pretty up the program. Change them to read "SOUTH" and/or "EAST" if that is where your home is located. The only convention that MUST be followed is to INPUT your longitude's degrees and minutes, EACH with a MINUS sign if your home is in the western hemisphere. If you live in the eastern hemisphere, longitude degrees and minutes are both positive.

ADDITIONAL TV SATELLITES:
May be added to the program program quite easily. Add line 341 IF NN = 16 THEN F = XX. XX would be the new bird's longitude. Plus, if in the eastern hemishpere, and minus in the western hemisphere.

Also add line 691 with the new bird's name, F;LO$,AZ,E,S and change lines 700 and 710 for < and = to 16. That's all there is to it.

NO LINE PRINTER?
No problem. Change line 700 to read: 700 INPUT RR: IF NN<15 GOTO 190. This will allow you to single step through each satellite's azimuth, range, and elevation. Just round off the output for each variable to the degree of accuracy desired and get out your paper and pencil hand-printer tools.

TYPICAL PRINTOUTS:
Pages 160 & 161 are typical line printer print-outs of the program. Page 160 is a print-out for a home in Loomis, California and Page 161 is for the author's QTH at Chautauqua Lake, New York. Most any road atlas contains more than adequate latitude and longitude info for you to determine your location's coordinates to within one minute accuracy.

On p.160, note that California cannot see STATSIONARs 4 and 8 because both are below the horizon, while having a good shot at STATSIONAR 10 if and when it is launched this year. Conversely, the author's location has a clear shot at both STATSIONAR 4 and 8, while STATSIONAR 10 is below the horizon.

SUMMARY:
Satellite TV "bird-watching" is one of the fastest growing hobbies within the hobby of amateur radio, today. The FCC's decision last fall '79 to eliminate mandantory licensing of TV receive only satellite terminals has given impetus to the proliferation of truly amateur (not-for-profit) terminals throughout the US. Satellite TV signals are not presently scrambled, but may well be in the not too distant future. When that time comes, the big "game" in town may indeed be played by advanced amateurs who learn to unscramble the signals as fast as some broadcasters encode them. A special "thank-you" to Ray Daly, for developing the fundamental part of the satellite location algorithm.

For those who wish to dig deeper into the subject of satellite TV we suggest they subscribe to the monthly journal, "Coop's Satellite Digest." It is expensive, but by far the best technical and programming (2 sections) journal available. It is available from: Satellite Television Technology, P. O. Box G, Arcadia, Oklahoma 73007.

```
100 REM W4UCH GEOSYNCHRONOUS SATELLITE AZ-EL PGM FOR TRS-80
110 CLS:NN=0:H=35800:PI=3.14159265:R=6367:CMD"T"
120 PRINT"STATION LATITUDE - ENTER DEGREES & MINUTES";:INPUTY1,Y2
130 PRINT"STATION LONGITUDE: USE + FOR EAST OR - FOR WEST":INPUTX1,X2
140 LPRINT"STATION LOCATION:":LPRINT
150 LPRINT"                LATITUDE =";Y1;"DEGREES";Y2;"MINUTES NORTH"
160 LPRINT:L1=X1*-1:L2=X2*-1
170 LPRINT"                LONGITUDE =";L1;"DEGREES";L2;"MINUTES WEST"
180 LPRINT:LPRINT:LPRINT:A=Y1+Y2/60:G=X1+X2/60
```

```
190 NN=NN+1
200 IFNN=1THENF=-135
210 IFNN=2THENF=-119
220 IFNN=3THENF=-132
230 IFNN=4THENF=-128
240 IFNN=5THENF=-95
250 IFNN=6THENF=-87
260 IFNN=7THENF=-99
270 IFNN=8THENF=-123.5
280 IFNN=9THENF=-91
290 IFNN=10THENF=-104
300 IFNN=11THENF=-109
310 IFNN=12THENF=-114
320 IFNN=13THENF=-14
330 IFNN=14THENF=-25
340 IFNN=15THENF=-170
350 CLS:B=G-F:IFB>180THENB=B-360
360 IFB<-180THENB=B+360
370 X=COS(B*PI/180)*COS(A*PI/180)
380 C=(180/PI)*(-ATN(X/SQR(-X*X+1))+1.5708)
390 IFABS(C)>=81.3THEN430
400 AZ=180+(180/PI)*ATN(TAN(B*PI/180)/SIN(A*PI/180))
410 IFA<0THENAZ=AZ-180
420 IFSGN(F)=-1THENLO$="WEST"ELSELO$="EAST"
430 PRINT"FOR A SATELLITE LOCATED AT";F;"DEGREES ";LO$:PRINT
440 IFABS(C)>=81.3THENAZ=0:F=F*-1:E=0:S=0:GOTO540
450 PRINT"AZIMUTH = ";AZ;"DEGREES":PRINT
460 AZ=INT(AZ*10+.5)/10
470 S=SQR(R↑2+(R+H)↑2-2*R*(R+H)*COS(C*PI/180))
480 PRINT"RANGE = ";S;"KM":S=INT(S):F=F*-1:PRINT
490 Y=(S↑2+R↑2-(R+H)↑2)/(2*R*S)
500 E=(180/PI)*(-ATN(Y/SQR(-Y*Y+1))+1.5708)-90
510 PRINT"ELEVATION = ";E;"DEGREES":E=INT(E*10+.5)/10
520 IFNN=1THENLPRINT"BIRD","LOCATION","AZIMUTH","ELEVATION","RANGE"
530 IFNN=1THENLPRINT"","DEGREES","DEGREES","DEGREES","KILOMETERS"
540 LPRINT
550 IFNN=1THENLPRINT"SATCOM 1",F;LO$,AZ,E,S
560 IFNN=2THENLPRINT"SATCOM 2",F;LO$,AZ,E,S
570 IFNN=3THENLPRINT"SATCOM 3 *",F;LO$,AZ,E,S
580 IFNN=4THENLPRINT"COMSTAR 1",F;LO$,AZ,E,S
590 IFNN=5THENLPRINT"COMSTAR 2",F;LO$,AZ,E,S
600 IFNN=6THENLPRINT"COMSTAR 3",F;LO$,AZ,E,S
610 IFNN=7THENLPRINT"WESTAR 1",F;LO$,AZ,E,S
620 IFNN=8THENLPRINT"WESTAR 2",F;LO$,AZ,E,S
630 IFNN=9THENLPRINT"WESTAR 3",F;LO$,AZ,E,S
640 IFNN=10THENLPRINT"ANIK 1",F;LO$,AZ,E,S
650 IFNN=11THENLPRINT"ANIK B",F;LO$,AZ,E,S
660 IFNN=12THENLPRINT"ANIK 3",F;LO$,AZ,E,S
670 IFNN=13THENLPRINT"STATSIONAR 4",F;LO$,AZ,E,S
680 IFNN=14THENLPRINT"STATSIONAR 8",F;LO$,AZ,E,S
690 IFNN=15THENLPRINT"STATSIONAR 10",F;LO$,AZ,E,S
700 IFNN<15GOTO190
710 IFNN=15THENLPRINT:LPRINT"* = GOOD LUCK RCA!"
720 LPRINT:LPRINT"NOTE: ZERO - AZ - EL - RANGE = NOT VISIBLE":END
```

RICHCRAFT ENGINEERING LTD.

Phone: (716) 753-2654

DRAWER 1065
CHAUTAUQUA, N. Y. 14722

LOOMIS, CALIFORNIA

STATION LOCATION:

LATITUDE = 38 DEGREES 50 MINUTES NORTH

LONGITUDE = 121 DEGREES 7 MINUTES WEST

BIRD	LOCATION DEGREES	AZIMUTH DEGREES	ELEVATION DEGREES	RANGE KILOMETERS
SATCOM 1	135 WEST	201.5	42.8	37583
SATCOM 2	119 WEST	176.6	45	37424
SATCOM 3 *	132 WEST	197	43.6	37521
COMSTAR 1	128 WEST	190.9	44.5	37461
COMSTAR 2	95 WEST	142	37.5	37987
COMSTAR 3	87 WEST	132.8	32.9	38370
WESTAR 1	99 WEST	147.1	39.5	37829
WESTAR 2	123.5 WEST	183.8	45	37425
WESTAR 3	91 WEST	137.2	35.3	38167
ANIK 1	104 WEST	153.8	41.6	37667
ANIK B	109 WEST	161.1	43.3	37545
ANIK 3	114 WEST	168.7	44.4	37463
STATSIONAR 4	14 WEST	0	0	0
STATSIONAR 8	25 WEST	0	0	0
STATSIONAR 10	170 WEST	241.3	22.8	39287

* = GOOD LUCK RCA!

NOTE: ZERO - AZ - EL - RANGE = NOT VISIBLE

RICHCRAFT ENGINEERING LTD.

Phone: (716) 753-2654

DRAWER 1065
CHAUTAUQUA, N. Y. 14722

CHAUTAUQUA LAKE, NEW YORK

STATION LOCATION:

LATITUDE = 42 DEGREES 12 MINUTES NORTH

LONGITUDE = 79 DEGREES 28 MINUTES WEST

BIRD	LOCATION DEGREES	AZIMUTH DEGREES	ELEVATION DEGREES	RANGE KILOMETERS
SATCOM 1	135 WEST	245.2	16.5	39918
SATCOM 2	119 WEST	230.9	27.1	38881
SATCOM 3 *	132 WEST	242.8	18.6	39706
COMSTAR 1	128 WEST	239.3	21.3	39435
COMSTAR 2	95 WEST	202.5	38.8	37885
COMSTAR 3	87 WEST	191.1	40.7	37739
WESTAR 1	99 WEST	207.8	37.4	37996
WESTAR 2	123.5 WEST	235.2	24.3	39148
WESTAR 3	91 WEST	196.9	39.9	37800
ANIK 1	104 WEST	214.2	35.3	38167
ANIK B	109 WEST	220.1	32.8	38373
ANIK 3	114 WEST	225.7	30.1	38612
STATSIONAR 4	14 WEST	107	9.3	40662
STATSIONAR 8	25 WEST	115.6	17.2	39842
STATSIONAR 10	170 WEST	0	0	0

* = GOOD LUCK RCA!

NOTE: ZERO - AZ - EL - RANGE = NOT VISIBLE

- CHAPTER 12 -

SELF-TEST QUESTIONS FOR CHAPTERS NOTED
(use decimal in all Chapts. where applicable)

CHAPTER 1:
1. What are the object codes for PUSH AF, BC, DE & HL?
2. What are the object codes for INCrement A,B,C,D,E,H, & L?
3. What are the object codes for PUSH IX and IY?
4. Which lines in page 17's program generates the hex code?
5. How does page 17's pgm. KNOW when the hex no. is completed?
6. What is the instruction for: 237 - 176, 237 - 86, 237 - 0?
7. What is the instruction for: 253 - 225, 253 - 203 - 46?
8. What are the instruction's' for following PEEK sequences?
   A. 62 - 63 - 6 - 123 - 144 - 255
   B. 225 - 213 - 193 - 241
   C. 253 - 237 - 221 - 253

CHAPTER 2:
Using ONLY your PEEK function, what are the instructions from the following Level II ROM locations? When a MEM location is called for by the instruction use this formula to determine the 2 byte address = PEEK (location +1) + 256 * PEEK (location + 2); i.e., PEEK(48) = 195 decimal = JP instruction.
Therefore, JP address = PEEK (48 + 1) + 256 * PEEK (48 + 2) = 16399 decimal = 400FH (for purists).
1. ROM locations 64 and 114 ? (all locations in decimal)
2. ROM locations 136 and 148 ?
3. ROM locations 155 and 159 ?
4. ROM locations 165 and 175 ?
5. ROM locations 178 and 192 ?
6. ROM locations 193 and 200 ?
7. ROM locations 201 and 204 ?
8. ROM locations 205 and 207 ?
9. ROM locations 209 and 217 ?
10. ROM locations 13 and 16 ?
11. ROM locations 22 and 38 ?
12. ROM locations 62 and 210 ?

CHAPTER 3:
1. When are integer operations output in single precision?
2. Using VARPTR what does L-3 output with integers?
3. Using VARPTR what does L-2 output with integers?
4. Using VARPTR what does L-1 output with integers?
5. Using VARPTR what does L output with integers?
6. Using VARPTR what does L+1 output with integers?
7. What NT (number type) does Level II RUN with unless instructed otherwise?
8. What is the NT for single precision? Double precision?
9. What is the NT for strings?
10. Why does CALL 260DH require considerable caution?

CHAPTER 4:

1. Using VARPTR what does L-3 output with strings?
2. Using VARPTR what does L-2 output with strings?
3. Using VARPTR what does L-1 output with strings?
4. Using VARPTR what does L output with strings?
5. Using VARPTR what does L+1 output with strings?
6. Using VARPTR what does L+2 output with strings?
7. What is the maximum length of a single string array?
8. How many bytes are CLEARed for $ by Level II on RUN?
9. How does one recapture auto-CLEARed string bytes?
10. Can the author think of a difficult string question?

CHAPTER 5:

1. Using VARPTR, what does K-3, K-2, K-1, K, K+1, K+2 and K+3 output with single precision numbers?
2. What is the largest power of 2 a TRS-80 can handle in single precision?
3. What BASIC function in ROM MAY introduce errors?
4. Which single precision byte holds the sign bit?
5. Which bit? How is it decoded?
6. How does Level II normalize negative numbers?
7. How does the program on page 102 determine the sign?
8. Page 101, line 50: what will be the value of EVERYTHING to the right of the * sign in this statement?
9. Page 104, pgm. WITHOUT USING EXPONENT SIGN: how are powers of 2 LESS THAN 1 derived?
10. Are single/double precision numbers a complex subject?

Chapter 6:

1. How does the pgm. on page 113 POKE 7D00H into the video display control block driver address?
2. IF you loaded this program in DOS, must you then load BASIC, SYSTEM, /32000?
3. IF you loaded this program in non-disk Level II, must you then type SYSTEM, program name, ENTER, load via cassette, and then BREAK after loaded?
4. Why is EX AF,AF' and EXX used instead of PUSH and POP?
5. What is the difference between LD DE,16 and LD E,16? Will the program work either way?
6. How does the author know to JP 0458H in line 520?
7. What is the line printer's "handshake" in decimal?
8. Where is it located in MEM?
9. How would you change this program to STOP printing with the BREAK key instead of the spacebar?
10. Will this program work with ALL NEWDOS+ EDTASSEM programs when the '123' modification is used?

CHAPTER 7:

1. Page 117: what is the function of the byte reserved by the label COUNT?
2. Why is PLACE DEFW rather than DEFB?
3. How could you eliminate line 240?
4. Why did not the author use SOME Level II BASIC CALLS instead of plain 'old Z-80 instructions?
5. Can the Z-80 ADD two 16 bit numbers ALL BY ITSELF?
6. If so, where is the SUM stored?
7. If so, what is the maximum value allowed?
8. Can this program be significantly shortened?
9. Can this program be easily relocated?
10. My dictionary says ELFIN is 'like an elf.' What is so ELFIN about this 59 line LONG program?

CHAPTER 8:

1. What are all the abbreviations on pages 120-122?
2. Without peeking at page 122, what are the decimal beginning and ending MEM locations of each video line?
3. Which takes the least EXECUTION TIME, a JR or a JP?
4. What is unique about a Cray computer?
5. Is this split screen program easily relocatable?
6. In a stand-alone assembly language program, how can you save 1, 2, or 3 16 bit words WITHOUT using MEM or stack in MEM?
7. How come this program uses a few ROM CALLS?
8. After loading HL, DE, & BC with appropriate values, can you insert a line BEFORE the opcode LDIR to DO something useful?
9. Does the author's program JUMP around a lot?
10. Can this program be significantly shortened?

CHAPTER 9 SUMMARY: no questions. NOTE: return to BASIC now = 114 decimal;F. added to COMPARE; DATA CONVER. last line added.

CHAPTER 10:

1. Why is TRS-80 keyboard relay K1 "NEVER USED" as a transmitter keying relay WITHOUT a buffer TTL chip and/or low voltage/low current buffer keying relay?
2. What is dot, dash, space between character elements, end of character, and end of word timing with relation to dot length?
3. In the program beginning on page 129: why are the variables A-Z defined as integers?
4. What does line 60 accomplish?
5. Why are lines 64 and 65 included?
6. Should a statement CMD"T" be included in line 58?
7. Could READ - DATA be used instead of lines 70-116?
8. If so, why were they NOT used?
9. What do the OUT 255s in line 124 accomplish?
10. How does the Morse code receive algorithm render the TRS-80's Z-24 flip-flop "invisible" to the incoming Morse signal?

CHAPTER 11:

1. Where can one locate the BEST reference article, 14 pages worth, on TV satellites? The BEST monthly journal?

2. What happened to RCA's SATCOM 3 bird last December '79?

3. How can you use this program if you are located in the Southern Hemisphere?

4. Should CMD"T" be removed from line 110 for non-disk systems?

5. What line determines if a given bird is NOT visible from your location?

6. Who put up the ANIK series of satellites?

7. Who put up, is putting up, the STATSIONAR series of satellites?

8. What frequencies do the U.S. satellites use to TRANSMIT the TV signals back to earth?

9. How many homes in the U.S. now receive some TV programs originally transmitted from source to satellite back to receive-only earth station, and then distributed locally via cable?

10. Is FCC authorization required to directly receive satellite TV programming?

NOTE:

Do not FORGET to enter BOTH longitude degrees AND minutes with a minus sign if your location is in the western hemisphere.

- CHAPTER 13 -

BIBLIOGRAPHY AND SELF-TEST ANSWERS

BIBLIOGRAPHY:

There are a number of very good texts that SUPPLEMENT Volumes 1 and 2 of the Disassembled Handbook For TRS-80. As our enlightened reviewer now perceives this plethora of worthwhile learning aids, they include:

BEST:
Dr. Lance Leventhal's "Z80 Assembly Language Programming," Osborne/McGraw Hill, 630 Bancroft, Berkeley, Cal. 94710, $9.95 postpaid. This is undoubtedly the best all-around reference book on the Z-80 for the advanced assembly language programmer.

BEST:
H. C. Pennington's "TRS-80 Disk And Other Mysteries," Computer Information Exchange, Box 158, San Luis Rey, Cal. 92068, $22.50 postpaid. Expensive, but a "must" for every disk user who wishes to know more about his DOS than just finding the on-off switch. This excellent book = The Disassembled Disk For TRS-80.

BEST:
Roger Fuller's "SUPERMAP," The Alternate Source, 1806 Ada Street, Lansing, Mich. 48910, $8.95 postpaid. Detailed memory map of Level II ROM. Undoubtedly, the BEST road map to virtually all significant Level II ROM addresses.

BEST:
Bi-monthly magazine for advanced TRS-80 programmers: The Alternate Source, $9.00 per year, 1806 Ada Street, Lansing, Mich. 48910.

BEST:
Anthology: "Pathways Through The Rom," Softside Publications, 6 South Street, Milford, N.H. 03055, $19.95 postpaid. This outstanding anthology includes Vol. 1 of the author's Disassembled Handbook, Roger Fuller's SUPERMAP, John Phillipp's HEX-MEM Monitor, and George Blank's Z-80 Disassembler. It is a BIG BOOK, and a must for those TRS-80 bookshelves that do not already have Vol. 1 and SUPERMAP.

BEST:
For the newcomer to Level II. Dr. David Lien's "Learning Level II," Compusoft Publishing, 8643 Navajo Road, San Diego, Cal. 92119, $17.40 postpaid. The very BEST Level II BASIC text for those who have completed the "User's Manual For Level I," by the same outstanding author. Highly recommended for ALL Level I graduates from ages 12 to 80, and from Jr. High School through post-doctoral levels.

GOOD:
Nathan Wadsworth's "Z-80 Software Gourmet Guide & Cookbook," Scelbi Publications, 20 Hurlbut Street, Elmwood, Conn. 06110, $15.95 postpaid. This cookbook's most redeeming facet is the well presented Chapter 8, "Floating Point Routines," for those who wish to dig deeper into the subject.

GOOD:
William Barden's "TRS-80 Assembly Language Programming," Radio Shack #62-2006, @ $3.95 per copy. On second thought, this book is indeed worth the price to the BEGINNING assembly language programmer as there is little else to choose from. There are virtually NO ROM CALLS used in this book, nor is the programmer ever told they exist, and may be used as desired. Had this book been built in Detroit, it would have been recalled, and the SIN of omission rectified at manufacturer's expense.

NOTE: Our peripatetic reviewer promises to return to his normal more candid habits in Volume 3 when the tranquilizers the publisher forced down his throat have 'worn off.'

- ANSWERS TO SELF-TEST QUESTIONS FOR CHAPTERS NOTED -

CHAPTER 1:
1. 245, 197, 213, 229
2. 60, 4, 12, 20, 28, 36, 44
3. 221-225 & 253-229
4. Lines 20 - 40.
5. IF E = 2 (line 45)
6. LDIR, IM 1, NONE
7. POP IY & SRA
8A. LD A,63 - LD B,123 - LD C,255
8B. POP HL, POP DE, POP BC, POP AF
8C. NONE = DATA list/table/constants, etc.

CHAPTER 2:
1. JP 1497 & JP 1644
2. LD (HL),44 & LD (HL),195
3. LD (HL),D & LD B,21
4. INC HL & CALL 7055
5. CALL 457 & RST 10H
6. OR A & LD A,H
7. OR L & LD A,(HL)
8. LD B,A & LD (HL),A
9. LD (HL),B & OR A
10. JP 1695 & JP 16387
11. JR to line 70 & JR to line 70
12. JR to line 27 & JR to line 199

Oh, Oh: questions 11 and 12 were 'dirty pool.' We threw in some JR's and never told you how to decode the relative jump. Here are the rules to this simple game:

A. IF the displacement byte AFTER the JR instruction is =< 127 THEN the jump MEM location = MEM location value of JR instruction + 2 PLUS value of displacement byte.

B. IF the displacement byte AFTER the JR instruction is => 128 THEN the jump MEM location = MEM location value of JR instruction + 1 MINUS complemented value of displacement byte.

Let's demonstrate these 2 rules with 2 examples on the next page. Remember, complementing a binary number is simply changing all the 1's to zeros & all the zeros to 1's.

(11A) PEEK (30) = 24 = JR instruction
(11B) PEEK (31) = 38
(11C) 38 =< 127 so 30 + 2 = 32 PLUS 38 = jump to line 70.

(12A) PEEK (54) = 24 = JR instruction
(12B) PEEK (55) = 227
(12C) 227 => 128 so complement 227 (all 1's = 0 & all 0 = 1)
227= 11100011 complemented = 00011100 = 28 decimal
54 + 1 = 55 MINUS 28 = jump to line 27.
The foregoing rules hold true for ANY kind of relative jump instruction including: JR, JR NC, JR NZ, and JR Z.

CHAPTER 3:
1. All divides & anytime output exceeds -32768 & + 32767.
2. Number Type (NT) = 2
3. Second ID of integer name or zero IF none.
4. 1st ltr. of INT name 5. LSB of integer in split-decimal
6. MSB of INT split-dec. 7. single precision
8. 4 and 8 9. 3
10. IF no variable found will create one and possibly MOVE previously located variables' addresses. USE CAUTION.

CHAPTER 4:
1. NT = 3 for strings 2. 2nd char. of $ name or zero
3. 1st char. $ name 4. string length
5. LSB string address 6. MSB string address in MEM
7. 240/255 bytes IF you CLEARed enough space
8. 50 9. CLEAR zero 10. NO such thing exists

CHAPTER 5:
1. NT, 2nd ID, 1st ID, LSB, NMSB, MSB, and exponent
2. Bottom page 97: 254 - 129 = 125
3. ↑ exponent function 4. MSB
5. first bit; 1 = minus and 0 = plus
6. same as positive numbers; sign bit of MSB determines sign
7. line 35: IF K => 128 THEN S = -1 to multiply result by
8. .5 to less than 1.0 9. Line 60: B/2 in FOR-NEXT loop
10. NO; they are absurdly simple when properly approached

CHAPTER 6:
1. Lines 160-180 2. No, it is self-initializing
3. Yes 4. saves 12 bytes MEM (incl stack)
5. 1 byte MEM; YES
6. It was already there in the Video Display Control Block's driver address in MEM at 4016H and 4017H.
7. 63 = ASCII "?" = "I'm ready for another char. handshake."
8. 37E8H = 14312 decimal in MEM
9. Change line 320 to: CP 4 (see page 139).
10. NO. Some EDTASSEM object code listings will foul it up, but most work ok. Page 114's object code was LPRINTed out using this feature as the SPACEBAR stop printing feature is a real convenience.

CHAPTER 7:

1. Frame counter MEM storage for frames 1 to 5.
2. DEFW = 2 bytes required for MEM address, LSB & MSB.
3. Initialize COUNT at ASCII zero = 48 decimal and modify balance of program where COUNT is used.
4. SOMETIMES a program can be written shorter, more concisely, use less MEM and RUN faster when only plain 'ole Z-80 instructions are used. Remember: SOMETIMES.
5. Sure it can; see lines 430 and 610.
6. HL register
7. 65535 = 1111111111111111binary
8. NO. Lots of luck if you wish to try.
9. YES. Changes would be to lines 120,130,150,160,170 & 660.
10. YES. The typesetter 'blew it.' It should real EL FIN = the END, in Don Quixote's language.

CHAPTER 8:

1. Ham radio has its abbreviations too. CQ = calling any station, DE = from, K = over to you/end of transmission, RST = readability/signal strength/tone quality with scale of 1 to 9, WX = weather, CD = civil defense, CW = Morse code/carrier wave, and OM = old man, but to name a few.
2. See page 122.
3. SURPRISE. JR takes 20% MORE execution time than JP. Look it up in the back of your Editor/Assembler Manual.
4. SPEED. Fastest commercial computer in the world. 70 MHz+
5. No, but not TOO difficult.
6. Use opcode EXX-put words in alt. regs & recall as desired
7. To save time & MEM; lines 35,120,150,190,245,260 & 765.
8. Sure you can. See line 105 that sets up ASCII SPACE.
9. Wait till he returns from delivering the bricks he is carrying to answer this question.
10. You bet it can. Try last Chapter's STORE & RESTORE.

CHAPTER 10:

1. It is rated at only 5 volts DC @ 500 milliamps MAX and will FAIL if used for larger currents.
2. Dot = 1 time interval, space between character elements = 1 dot, dash = 3 * dot, end of character = 3 * dot, and end of word = 7 * dot.
3. Program execution speed as integers are handled MUCH FASTER. After Chapters 3, 4, & 5 you SHOULD know it.
4. Sets the TRANSMIT code speed.
5. They allow a DELIBERATELY induced error such as BREAK, @, then ENTER to immediately return us to TRANSMIT mode.
6. Yes, when used with disk to speed-up execution time.
7. YES.
8. Much, much slower execution time.
9. Close & open cassette motor control relay at Morse rate.
10. By resetting the Z-24 flip-flop EVERYTIME a time measurement is made. See lines 179, 184, and 189.

CHAPTER 11:

1. 73 Magazine, November '79, pp 120 - 133 (most libraries have it and/or most radio amateurs subscribe). Satellite Television Technology, P. O. Box G, Arcadia, Okla. 73007. It is composed of 2 sections: Technical & Programming. Either section is available at $30/year or both sections at $50/year...it is a monthly periodical to which this author contributes a monthly column: "The World Above 10 GHz."

2. It probably "blew-up" during the burn to move the bird from holding orbit into proper equatorial geostationary orbit, early December '79....appproximately $70 million "firecracker."

3. Very easily. Just change Line 150 to read: "MINUTES SOUTH" instead of "MINUTES NORTH." The program could care less as azimuth and elevation to a given EQUATORIAL bird are identical for locations with the same latitude and longitude whether NORTH or SOUTH. Try it.

4. Yes.

5. Line 390

6. Canada

7. Russia

8. 3.7 to 4.2 GHz = 3700 MHz to 4200 MHz.

9. Around 14 million as of May 1980.

10. No; not since fall '79....BUT, you WILL have a copyright/ legal problem with the satellite owners/operators IF you "sell it to neighbors." It is little different than xeroxing this copyrighted book and selling it on a street corner. You will probably 'get away' with the latter, but it is difficult if not impossible to 'hide' a 10' or 12' diameter parabolic reflector antenna that is necessary to receive the TV satellites' signals. Good luck.

CHAPTER 14

- APPENDIX -

DISK/CASSETTE PROGRAMS FROM VOLUMES 1 & 2:

Richcraft has received a number of inquiries from computer science departments and individuals regarding availability of the copyrighted programs in Volumes 1 and 2. As such, we prevailed upon a professional cassette/disk duplicator to make them available. They are available to BOTH individuals and schools at $20. cassette or 35/40 track disk postpaid, for ALL programs listed below from:

The Alternate Source
1806 Ada Street
Lansing, Michigan 48910

Richcraft Engineering Ltd.
Drawer 1065
Chautauqua, New York 14722

PROGRAM COPYRIGHT (c) 1980	FILE NAME:
Basic Functions Name List/BAS	A
Basic Functions Call Adresses/BAS	B
Integer Arithmetic - Source Code w/comments	C
Integer Arithmetic - Object Code	D
Single Precision Arithmetic - Source Code w/comments	E
Single Precision Arithmetic - Object Code	F
Double Precision Arithmetic - Source Code w/comments	G
Double Precision Arithmetic - Object Code	H
ROM Function Demonstration - Source Code w/comments	I
ROM Function Demonstration - Object Code	J
Multi-Base Number Conversion/BAS	K
LPRINT All Zeros With Slash - Source Code w/comments	L
LPRINT All Zeros With Slash - Object Code	M
Single Precision Decoding/BAS	N
Single Precision Decoding (no exponent sign)/BAS	O
Double Precision Decoding/BAS	P
Double Precision Decoding (no exponent sign)/BAS	Q
'JKL' LPRINT Out Video - Source Code w/comments	R
'JKL' LPRINT Out Video - Object Code	S
Storing Video In MEM - Source Code w/comments	T
Storing Video In MEM - Object Code	U
Split Screen Video - Source Code w/comments	V
Split Screen Video - Object Code	W
W4UCH Morse Code Transmit & Receive/BAS - Part 1	X
W4UCH Morse Code Transmit & Receive/BAS - Part 2	Y
TV Satellite Locator/BAS	Z

NOTE:

All above programs are copyrighted and sold under license from Richcraft Engineering Ltd.

INDEX FOR VOLUMES 1 AND 2

- A -

ACCUM demonstration exercise 2-109
ACCUM demonstration text 2-108
alphabetical CALL summary decimal 1- 68 2-138
alphabetical CALL summary hex 1- 35
ancillary function summary 2-139
ancillary ROM subroutines 1- 29
arithmetic CALL summary 1- 32 2-140

- B -

BASIC function CALLS hex & decimal 1- 12
bibliography 1- 64 2-166

- C -

compare table 1- 32 2-140
conversions, data 1- 32 2-140

- D -

data conversion 1- 32 2-140
data movement table 1- 31 2-139
data movement text 1- 30
decoding double precision 2-104
decoding single precison 2- 95
disassembled ROM - partial 2- 21
disassembling TRS-80 object code 2- 8

- E -

exponent 'end run' program 2-102
exponent problem - program 2- 98

- F -

function name list - program 1- 7
function Call address - program 1 8

- I -

integer arithmetic program 1- 17
integer arithmetic text 1- 14
integers, decoding 2- 84

- J -

'JKL' LPRINT video program 2-113
'JKL' LPRINT video text - '123' optional 2-111

- K -

keyboard MEM locations and values 1- 29 2-139
keyboard Call routines 1- 30

- L -

line printer text 1- 33
LPRINT all zeros with a slash - source 1- 56
LPRINT all zeros with a slash - object 1- 57

- M -

Morse code program - part 1 2-147
Morse code program - part 2 2-149
Morse code program text 2-141

- N -

normalization 2-100
number conversion program, multi-base 1- 53
number conversion program text 1- 11

- O -

object codes, instructions numerical order 2-11
object codes, Level II ROM - partial 1- 37

- P -

print all zeros with a slash, object code 1- 57
print all zeros with a slash, source code 1- 56

- R -

ROM Calls, decimal 1- 68 2-138
ROM Calls, hex and decimal 1- 12
ROM Calls, summary 1- 35

- S -

Self-Test answers 1- 66 2-167
Self-Test questions 1- 58 2-162
single precision arithmetic program 1- 19
single precision arithmetic text 1 -16
single precision, decoding 2- 95
split screen displays 2-120
split screen video program 2-125
split screen text 2-119
storing video in MEM program 2-117
storing video in MEM text 2-115
summary 1- 5 2- 4

- T -

TV satellite locator printouts 2-160
TV satellite locator program 2-159
TV satellite locator text 2-157

- U -

Using ROM subroutines, Call list 1- 26
Using ROM subroutines program 1- 27
Using ROM subroutines text 1- 23

- V -

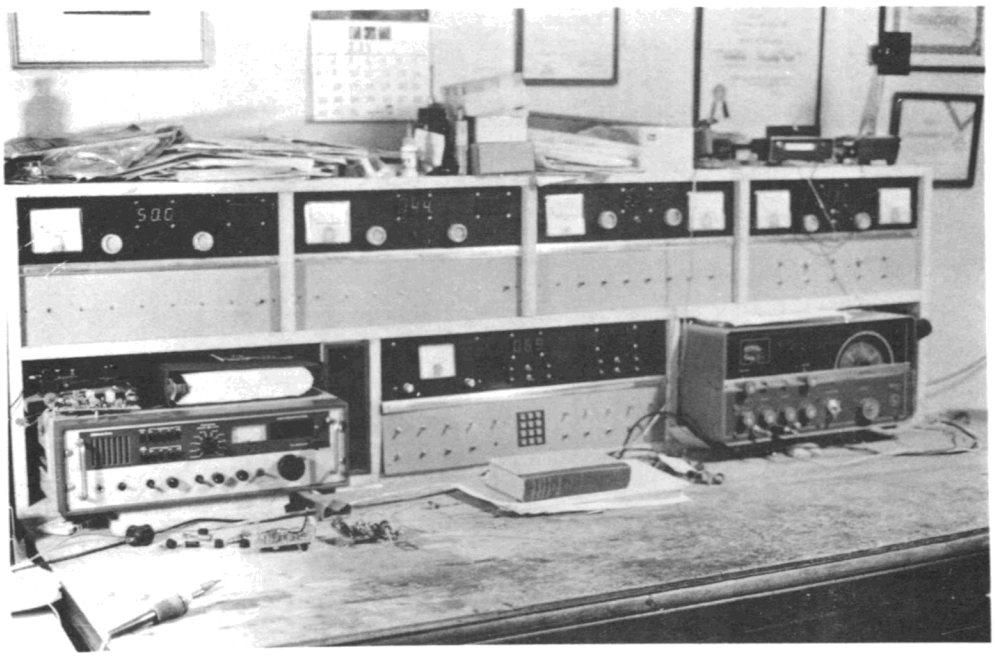
VARPTR, integers programs 2- 84
VARPTR, integers text 2- 84
VARPTR, double precision programs 2-104
VARPTR, double precision text 2-103
VARPTR, single precision programs 2- 97
VARPTR, single precision text 2- 95
VARPTR, strings and string arrays programs 2- 91
VARPTR, strings and string arrays text 2- 91
video display MEM locations, decimal 2-122
video display text 1- 33

- Z -

Z-80 instruction set 2- 11

- AUTHOR'S TRS-80 WORK STATION -

TOP ROW CENTER : 2ND VIDEO DISPLAY FOR COMBO T.V. READOUT
NEXT ROW : 1ST VIDEO DISPLAY & EXPANSION INTERFACE
TABLE ROW : WESTERN I/O IBM PRINTER, KYBD, & CASSETTES
BENEATH TABLE RT. : DUAL PERCOM/WANGCO 82 MINI-FLOPPY DISKS



- AUTHOR'S AMATEUR RADIO STATION W4UCH/2 -

TOP FAR RT : GUNNPLEXER 10 GHz TRANSCEIVER - VIDEO & AUDIO
ABOVE DESK : VHF TRANSVERTERS 50 MHz-144 MHz-220 MHz-432 MHz
DESK LEVEL : ITT 3021 DIGITAL RECVR, CONSOLE ANT/TIME, HT-37
BELOW DESK : REGULATED POWER SUPPLIES