A PILOT LANGUAGE SYSTEM

by

LARRY TRISTAN WALKER

B. S., Georgia Institute of Technology, 1961

A MASTER´S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1978

Approved by:

Myron A. Calhoun
Major Professor

TABLE OF CONTENTS

INDEX OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 PURPOSE OF THE REPORT

The purpose of this report is to describe the implementation of a microprocessor-based Computer Assisted Instruction (CAI) system for the PILOT computer language. The report provides documentation sufficient to support software maintenance and user guides for the system. PILOT stands for Programmed Inquiry, Learning or Teaching and is a relatively easy to learn computer language. This report presents an outline of the general requirements of a CAI system and rational for selecting the PILOT language for this microprocessor-based implementation.

1.2 HISTORY OF PILOT

A principal barrier to the use of CAI has been the large amount of time needed to write courses in a general purpose language such as FORTRAN. The effective teaching of a majority of subjects requires the presentation of large amounts of text and the receipt of student answers in words.

The programming languages that have significant text-processing features, like SNOBOL, appear to be well suited for CAI. These languages introduce another problem, however, as the languages are too complicated for most teachers to learn; and therefore a computer programmer is required to prepare the course software (courseware).

Languages have been developed for writing courseware. Most of them, like COURSEWRITER, PLANIT, and TUTOR are large, complex languages that require a major effort to learn. This author's experience using PLANIT at the U.S. Army Command and General Staff College, Fort Leavenworth, Kansas, was that it took twenty to thirty hours of intensive study (via CAI) for nonprogrammer graduate students to develop a rudimentary PLANIT programming skill.

In approximately 1970, John Starkweather of the University of California Medical Center developed PILOT. It was significantly different from the CAI languages seen before as it was exceptionally simple in form. It was used at Stanford Research Institute in an education research study. Findings of this study were that teachers and students learned the language very easily and were able to write programs immediately [1].

The National Library of Medicine, Bethesda, Maryland has been a strong supporter of PILOT. Through a grant from that agency, John A. Starkweather developed an assembly language version of PILOT for the Intel 8080 microprocessor [2]. This version, known as 8080 PILOT, was the starting point for this project.

1.3 SYSTEM ORGANIZATION

The system described by this report consists of an adaptation of 8080 PILOT to a specific hardware grouping, integrated with system level software to provide for file creation, storage and retrieval. The software is divided into a PILOT author subdivision, a student subdivision and an operating system subdivision which supports both the PILOT author and student elements. Maximum practical use is made of existing software modules to perform functions of the overall system.

The hardware elements consist of a Digital Group Z-80 microcomputer system located in the Microprocessor Laboratory of the Department of Computer Science at Kansas State University. Input devices into the hardware configuration consist of keyboard, digital cassette tape and audio cassette tape. Output devices consist of CRT display, impact printer, digital and audio tape cassette.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 CLASSES OF USERS

CAI systems typically have three general classes of users. The class with the largest population is the student class. This class is characterized by having little computer experience and limited ability to follow complex instructions. Of course there are many exceptions to this profile but the developer of a CAI system should plan to support a student with these basic characteristics.

The next class of users is the CAI author class. They will be the creators of courseware (lesson material) for the system. The span of computer qualification of this class is great. The utility of the system can be greatly enhanced if it can be used by those among the lowest segment of qualification.

The last class of user is the system maintainer. Functions for this user include updates to software, backup of mass storage, purging of unwanted files and other duties as desired. He needs the widest possible scope of capabilities.

2.2 FUNCTIONAL PERFORMANCE

The student's use of the system is limited to lesson execution. In this mode he requires rapid execution to remain interactive with the system and a simple control structure to avoid complex instructions. For example, the system should be very simple to start up, load a lesson and begin execution. It is important to note that execution should be rapid while in the lesson, i.e., no frequent long pauses.

It is this author's observation that the student is very tolerant to a minutes pause after five or ten minutes of intensive interaction. This tolerance is very high if the lesson material conducts a dialog to explain the delay, i.e., "catch your breath while I load a new lesson for you." This concept for user tolerance is a critical factor in the system configuration.

The CAI author use of the system includes lesson preparation, trial lesson execution, and lesson file storage and retrieval. The author peformance speed requirements are similar to that of the student. He expects rapid response while in the lesson creation mode (which is, in reality, an editor environment). For text changes, he expects rapid response. His tolerance for delay at the end of a lengthy editing session is not dissimilar to the student tolerance for pauses between lessons.

The maintainer can be considered as a user that could employ the most rapid response that can be achieved within the hardware limits of the system.

The very nature of CAI requires that lessons be transported easily from one system to another via a compact, inexpensive and reliable media. The above general enumeration of user requirements dictates the system performance.

2.3 HARDWARE CONSIDERATIONS

User requirements for system performance determine the extent of hardware in general. The CAI environment typically requires rapid execution response with reasonable toleration for infrequent pauses as outlined in preceeding paragraphs.

Cost can become a principal consideration. In a Z-80 system built of The Digital Group Inc. [3] components, the following price structure is applicable: Floppy Disk Drive (Single) System, 240,000 bytes of storage, $1,195.00; Digital Cassette (Dual) Storage System, 1,300,000 bytes of storage, $675.00; Disk Storage Media $7.00; Digital Cassette Storage Media $1.80. It is significant to note that the lower cost cassette-based mass storage systems may satisfy the user's requirements as well as the higher cost disk mass storage system, since the user has a tolerance level for much longer file loading time of the digital cassette system. In a CAI environment this file loading occurs at lesson change time for the student and edit begin/end time for the CAI author. If courses are long, the cassette advantage increases as it provides for greater storage at a much lower media cost and requires less frequent changes of media to get additional lessons.

CHAPTER 3

HARDWARE IMPLEMENTATION ALTERNATIVES

3.1 HARDWARE CONFIGURATIONS AVAILABLE

The Kansas State Department of Computer Science has two microprocessor systems that are candidates for the 8080 PILOT implementation. These are the ALTAIR 8800 (Intel 8080 microprocessor) and the Digital Group (Z-80 microprocessor) systems.

The ALTAIR class of system is exactly the system upon which the 8080 PILOT was designed to be implemented. The 8080 PILOT is based on lesson loading from asynchronous paper tape via program command and character by character output of lesson material via teletype. The Digital Group Systems, on the other hand, do not have any asynchronous loading capability and the prinicipal output media is a video display system that requires far more complex logic than the simple teletype protocol the 8080 PILOT is designed to support.

3.2 SELECTION CRITERIA

The major strength of the Digital Group hardware is that it has computer-controlled mass storage media and this single feature prompted its selection as this author desired to create a practical CAI system. Mass storage under computer control, in the opinion of this author, is required for a practical CAI system.

Further considerations were the factors of ease of start up of the two systems. The ALTAIR requires the bit by bit switch entry of a binary bootstrap program while the other system requires only pushing the "on" button as The Digital Group hardware has a Read Only Memory (ROM) start up facility. A final consideration was the fact that this author is thoroughly familiar with the Digital Group System as he owns one himself!

CHAPTER 4

SOFTWARE IMPLEMENTATION ALTERNATIVES

4.1 TOTAL SYSTEM SOFTWARE REQUIREMENTS

The student software facility requires a method of loading the PILOT interpreter initially, loading lessons initially and executing the lesson material. While executing, the student facility may require the ability to branch to a new lesson. This feature requires automatic loading of the desired lesson and execution of that lesson. Note, this automatic loading and subsequent execution is not possible under the original 8080 PILOT as it has no facility for control of processor-managed mass storage. Therefore, the automatic branching to new lessons represents a major enhancement to 8080 PILOT.

The PILOT author software facility requires all of the student capability for the trial execution of lessons and, in addition, requires the lesson creation ability via a lesson editor program. Again the 8080 PILOT does not have an editor and thus this lesson editor capability represents another major enhancement to 8080 PILOT. The PILOT author also requires the ability to move lesson files to and from the editor to mass storage at the start and end of lesson edit sessions.

The maintainer requires all of the PILOT author facility. In addition, the maintainer requires the ability to change the code of the PILOT interpreter, to debug programs, to manipulate files on a group basis and to prepare backups of the mass storage media.

4.2 SOFTWARE SOURCES

The 8080 PILOT interpreter was available only as an Intel 8080 assembly listing and could not be obtained in any machine readable media [4]. This listing did include a hex listing of the machine language.

An acceptable editor was available as a standard product of The Digital Group, Inc. This editor is an adaptation of an editor previously published in Dr. Dobb's Journal [5]. No listing was available of the Digital Group Editor, and it has been extensively modified from the original version. A listing was available of the original published version.

The standard product Digital Group Operation System, PHIMON (Phideck monitor) was available to provide the required system level support and maintenance capabilities required for a complete system. A BASIC interpreter, MAXI BASIC 1.1, was available. This is a standard Digital Group Software product and has extensive display interface routines needed to adapt 8080 PILOT to the Digital Group display. No documentation was available of the internal operations of MAXI BASIC.

4.3 AVAILABLE SOFTWARE DEVELOPMENT TOOLS

Since the 8080 PILOT was available in Intel 8080 assembly language, the principal software tool needed was an Intel 8080 assembler. Unfortunately, this critical item was missing. Only a Zilog Z-80 assembler was available. While the two instruction sets are compatible, the mnemonics for the operation codes and assembler formats are drastically different. It has been this author's experience that translation from one set of instructions to the other is a very error prone operation.

In addition, a Zilog Z-80 symbolic disassembler was available along with excellent facilities for direct entry of code into memory and dumping of memory contents to display and printer.

4.4 SELECTED SOFTWARE TECHNIQUES

A decision was made to enter the 8080 PILOT code directly into memory in machine language, dump this code in hex format, manually compare the dump with the original listing for errors and modify the code at the machine language level. The Zilog Z-80 disassembler was used to document the resultant end product. This process was facilitated by having a skilled typist enter the original listing. Very few errors were discovered in the entire process. Those that were discovered have been, in the majority, uncovered by the disassembler producing nonsense code because the errors were not logical instruction seqúences.

In cases where major changes were made to the 8080 PILOT interpreter, the new code was assembled with the Zilog Z-80 assembler and patched into the main body of machine language code. All changes made to the editor were made at the machine language level.

The display routines were developed by disassembling the MAXI BASIC routines in their original location, moving the code in memory to the PILOT machine language segment, and then correcting all nonrelative addresses and external calls.

4.5 CONCLUSION

The use of machine language level programming is a difficult process and in this project has increased the time required to complete the PILOT segment of the software. With the clarity of hindsight, the author observes that time necessary to implement an Intel 8080 assembler would probably have been recouped in saving of time spent on machine language programming. The greatest advantage to be achieved, however, would lie in increases in the ease of maintenance of the PILOT interpreter. If the 8080 PILOT source code had been available in machine readable media, the machine language approach would never have been taken.

CHAPTER 5

PILOT LANGUAGE PROCESSOR DESCRIPTION

5.1 SYSTEM LEVEL ORGANIZATION

Comments in this chapter pertain to the original 8080 PILOT interpreter by Starkweather. The reader is encouraged to refer to Appendix A, PILOT Language Definition, and reference 4, "Source Code for 8080 PILOT," as required for comprehension. The original PILOT interpreter was organized into 72 assembly language routines and storage areas. The interpreter provided for the entry of a PILOT lesson but with no provision for editing that lesson or even adding new material to the end of a lesson.

Printing, displaying, storing (on paper tape) and execution of lessons was provided. The command structure of the interpreter provided for branching to an editor but no editor was included. Provisions were also made to allow branching to a BASIC interpreter for statement evaluation but no such interpreter was provided. A facility was provided for loading a new lesson from paper tape.

5.2 PROCESSOR OPERATIONS

A typical session with PILOT starts by utilizing the "Input" routine (0CCFH of reference 4) to place a lesson in the program buffer. Once this is complete, the user can either execute this lesson or store it on paper tape. If a choice is made to store it, control passes to the "save" routine (0C13H of reference 4).

Subsequent execution of this lesson turns control over to "Interpret Existing Program, IEP (057CH of reference 4). Execution follows a logical path of Scan Input Buffer, SCAN (0380H of reference 4); Interpret Operation Code, OPS (03CEH of reference 4) and return to SCAN for more code. Figure 5.1 illustrates this general flow. Interpret Operation Code calls other elements of the interpreter as required.

5.3 MEMORY UTILIZATION

The 8080 PILOT interpreter resides from 0080H to 0EEAH with lesson material stored from 1000H upward. String variables are stored from 1FFFH downward. The code from 02E0H to 0EEA is never modified and could be placed in Read Only Memory (ROM) if desired.

5.4 INPUT AND OUTPUT OPERATIONS

The code is remarkably well organized for input/output. A jump table (located at 0100H) defines all the locations for I/O routines. This table organization gives great flexibility to change I/O even if the interpreter is placed in ROM.

I/O capabilities include asynchronous character-by-character interface to an ASR Model 33

```
       *********
      * STARTUP *
       *********
           !
!--------->!<--------------------------------------------
!          !                                            !
!          *                                            !
!         * *                                           !
!        *   *                                          !
!       *     *                                         !
!      * LOAD  *   NO                                   !
!     *  A NEW  *-----------------------!               !
!      *PROGRAM*                        !               !
!       *  ?  *                         !               1
!        *   *                    *************         !
!         * *                     * INTERPRET *         !
!          *                      * EXISTING  *         !
!          ! YES                  *  PROGRAM  *         !
!          !                      *************         !
!          V                            !               !
!     ***********           !---------->!               !
!     *  INPUT  *           !           !               !
!     * ROUTINE *           !           V               !
!     ***********           !    **************         !
!          !                !    * SCAN INPUT *         !
!          !                !    *   BUFFER   *         !
!          !                !    **************         !
!          V                !           !               !
!          *                !           !               !
!         * *               !           V               !
!        *   *              !     *************         !
!       *     *             !     * INTERPRET *         !
!  NO  * SAVE  *            !     * OPERATION *         !
!<----* LESSON  *           !     *    CODE   *         !
!      *   ?   *            !     *************         !
!       *     *             !           !               !
!        *   *              !           V               !
!         * *               !           *               !
!          *                !          * *              !
!          !                !         *   *             !
!          ! YES            !        *     *            !
!          V                !  NO   *  END  *           !
!    ****************       !------*    OF   *          !
!    * SAVE ROUTINE *               * LESSON*           !
!    * (Punch Paper *                *     *            !
!    *    tape)     *                 *   *             !
!    ****************                  * *              !
!          !                            *               !
!          !                            !               !
!          !                            !  YES          !
!----------!                            !---------------!
```

FIGURE 5.1 PILOT GENERAL FLOW

Teletype, jump to Editor (null condition, returned to monitor), jump to BASIC (null condition, returned with no action) and jump to a monitor. This monitor was not part of the 8080 PILOT implementation but was assumed to reside in the system in high memory.

CHAPTER 6

PILOT LANGUAGE PROCESSOR MODIFICATIONS

6.1 SYSTEM LEVEL REQUIREMENTS

The major changes required to the PILOT processor involved integrating it into an operating system structure to provide for lesson files loading, interface to the keyboard and display system of the Digital Group Z-80 microprocessor system, and changing the internal operation of the processor to provide for automatic lesson execution on initial start up. This last feature was deemed necessary for creation of a very simple student environment.

6.2 DISPLAY DRIVER

The source of the display driver code was the Digital Group MAXI BASIC interpreter. It appeared ideally suited to interfacing a character-by-character output into the Digital Group Display. The MAXI BASIC routine was first disassembled to determine external routines that it addressed. It was found that it addressed only the "Character Out" and "Space Out" routines of the Digital Group standard ROM. Since PILOT uses the standard ROM area for stack space, it was not possible to use these "Character Out" and "Space Out" routines. Similar routines exist in PHIMON and references to these routines were substituted for the ROM routines. This created a small constraint. PHIMON must be resident in the system for the Display Driver to work. Calls to "Character Out" (E6B2H) were inserted at 0F79H, 0F84H, and 0FHFH (see Appendix C for symbolic code listing).

The display buffer for the screen requires 1024 decimal bytes. Memory locations 1000H through 13FFH are used for this purpose. Nonrelative addresses of all the elements of the code were identified, the code was moved to 0F00H with a small memory move program, and all nonrelative addresses were adjusted for the new location.

6.3 KEYBOARD DRIVER

The source of the keyboard driver was the MAXI BASIC interpreter. The MAXI BASIC keyboard routine is contiguous with the MAXI BASIC display routine. This keyboard routine was moved and had nonrelative addresses adjusted concurrently with the display driver routine as described in Section 6.2 In addition, a subroutine call was changed as described below. The keyboard driver called a "Keystroke In" routine located in low memory of the MAXI BASIC interpreter. This call (located in the keyboard driver at OFB2H) was changed to call a "Keystroke In" routine (located at E36AH) of PHIMON. The use or the PHIMON "Keystroke In" routine eliminated the requirement to include a "Keystroke In" routine in PILOT.

6.4 FILES INTERFACE

The files interface required the creation of a linkage to allow file commands for loading new lessons to be passed from the PILOT lesson under execution to the PHIMON operating system and, if required, error messages to be passed back to the user. The technique used was to dynamically modify a segment of PHIMON code to change it from a general purpose overlay to a large subroutine to accomplish the specific file loading required. The modified code was altered so that it no longer appeared as a correct system overlay. This alteration prevented PHIMON attempting to use the modified code as a general purpose overlay.

In addition, formatting was performed on the PILOT command to convert it to the form needed by PHIMON. The formatted command was then moved from the PILOT input buffer to PHIMON's command buffer.

The Z-80 Assembler was used to produce this code. A message editor that resides in PHIMON was used to output error messages. A detailed, commented listing of the code is in Appendix D, Assembly Language Listing of Files Interface for Z-80 PILOT. This code is envoked by a call located at 0BE7H of the PILOT interpreter.

The PILOT command format for loading new lessons from deck 0 is:

```
"LOAD: FILENAME"
```

(where FILENAME is any legal PHIMON filename). The command for loading from other tape decks is:

"LOAD:#X FILENAME"

(where X is the drive number). If an error occurs, the logical sequence of the lesson is broken, so the routine gives the student the error message:

"PILOT LESSON DESIRED NOT ON THIS TAPE. ASK FOR HELP."

The PILOT system is then halted and control is given to PHIMON.

6.5 AUTOMATIC START UP

Originally PILOT expected the user to start the system into executing the lesson or preparing a new lesson. If the alternative of preparing a new lesson was selected, the system immediately cleared the lesson (program) buffer of all its contents and prompted the user to enter a new lesson from the keyboard. In summary, PILOT started up under control of the student, and unless the student proceeded properly, the lesson prepared by the author would never gain control of the teaching environment.

This new implementation of PILOT has changed that sequence so that the PILOT system initially starts up under control of the lesson rather than under control of the student. The student merely has to push a button to turn the system on and then enter:

"RUN PILOT" (carriage return)

The system starts up under control of the author prepared lesson.

This modification was accomplished by moving the start up buffer location from the input buffer (0E72H of reference 4) to the program buffer, the location where PILOT lessons are loaded. This creates one potential problem. If PILOT is run without any lesson loaded, it will attempt to execute whatever is in memory in the lesson storage area. This is simple to overcome by always having some small lesson loaded as part of the PILOT system. Once the details of the PILOT interpreter were understood, this modified start up procedure was simple to accomplish. The first line of the "Initialize Data for New Program" routine (0356H of reference 4) was changed from:

```
INIT LX1 H, IBUFF
       TO
INIT LX1 H, PBUFF.
```

In addition, in this same "INIT" routine the calls to "NEWN" and "INITV" were eliminated. This retains all variables from one lesson to the next. The variables can be cleared if desired by the incoming lesson so all alternatives are possible under this arrangement.

6.6 MEMORY UTILIZATION

A map of memory for the new PILOT interpreter is contained in Figure 6.1. Digital Group Systems operating under PHIMON require memory from E000H to E7FFH for PHIMON root code and overlays. A ROM is located from 0000H to 00FFH for dead-starting the system and for limited display

```
E7FFH   ----------------------------------------------
        !
        !   PHIMON O/S  (800H BYTES)
        !
DFFFH   !---------------------------------------------
        !                                          !
        !            STRING VARIABLE STORAGE       !
        !   (Note 1)                               ↓
        !      ↑     PILOT LESSON STORAGE
        !      !
14FF    !---------------------------------------------
        !
        !   FILES I/O INTERFACE  (100H BYTES)
        !
13FF    !---------------------------------------------
        !
        !   DISPLAY BUFFER  (400H BYTES)
        !
0FFF    !---------------------------------------------
        !
        !   DISPLAY & KEYBOARD DRIVER  (100H BYTES)
        !
0F00    !---------------------------------------------
        !
        !   PILOT INTERPRETER (C20H BYTES)
        !
02E0H   !---------------------------------------------
        !
        !   SYSTEM DATA (1C8H BYTES)
        !
0118H   !---------------------------------------------
        !
        !   JUMP TABLE  (18H BYTES)
        !
0100H   !----------------------------------------------
        !         !
        !   STACK ↓
        !
        !   AREA BELOW 0080H NOT USED BY PILOT
        !
0000H   !----------------------------------------------
```

Note 1: Upper limit is a function of available memory up to a maximum of DFFFH. This maximum provides 51,968 decimal bytes of lesson storage.

FIGURE 6.1 Z80 ¦LOT MEMORY ALLOCATION

I/O routines.

PILOT requires that the ROM be disabled prior to running because PILOT uses the top of the ROM area for stack. This configuration for PILOT will operate in a system as small as 8K bytes (counting the high memory PHIMON space). This would leave only 768 decimal bytes for lesson and string variable storage. In a 10K system, 2816 decimal bytes would be available for lesson and string variable storage. This 10K system represents the minimum level memory that the author recommends.

Adapting PILOT to various size memory configurations is very simple. Bytes 0360H and 0361H (low address, high address) should be set to the top of the PILOT lesson storage address.

CHAPTER 7

EDITOR MODIFICATIONS

7.1 SYSTEM LEVEL REQUIREMENTS

The editor is to be used only by the author in lesson creation and modification. The lesson preparation task is much like any other program creation task and no unusual edit features are required. The Digital Group Text Editor is typical of an editor designed for small systems with the most typical use being the creation of source files for assemblers. Detail description of the capabilities are outlined in Appendix G, PILOT Author System Users Manual.

The editor must be able to create files in memory, to provide for interface to the file storage system for saving the files, and to accept files from the system for modification. The Digital Group Editor can do all these operations, however, the files interface requires the user to enter simple OS job control language, i.e., "SAVE MYFILE XX-YY," where XX-YY are memory locations.

7.2 MEMORY UTILIZATION

The editor originally used 0100H through 1400H for its own code and 1400H upward for edit files. If used for PILOT lesson creation, it would be possible to prepare a lesson which would overlay the PILOT routines and create a totally unworkable system. This could be remedied by changing the control parameters of the editor to ensure all edit files are created above 14FFH.

The problem was to determine where the minimum control file parameter was located in the editor. This problem was compounded by the fact that only an assembly language listing for the code that the assembler had been adapted from was available. The correct location of the control value was finally located by the implementation of a searching routine to locate the value to control minimum edit file memory. Had this routine been created earlier, it would have helped greatly in making machine language changes to PILOT code.

7.3 PRINTER DRIVER

The editor had originally been configured to drive a teleprinter type device with 110 baud ASCII. The hardware that was used to implement PILOT has a Digital Group Printer, a matrix printer operating under extensive software control with even the character set defined by a pattern held in memory. In addition, the Digital Group Printer routines, as supplied by Digital Group, had an error condition which produced an extraneous character at the end of each line of print. A Z-80 assembly language listing of this code was available so the code was typed into the Z-80 assembler, corrections made and assembled into the hard copy routine space of the editor. The end product was a fully functional interface between the editor and the Digital Group Printer without errors in printing.

7.4 CAPABILITIES TAILORING

The original editor had a "software front panel" which permitted the user to view all CPU registers, and memory set any memory location to any value, and dump or load any portion of memory to or from audio tape. These capabilities were not required by the author user and therefore the entries to these routines were eliminated. The remaining capabilities are:

1. EDIT PILOT LESSON - DISPLAY
2. EDIT PILOT LESSON - HARDCOPY
3. RETURN TO PHIMON

CHAPTER 8

CONCLUSIONS

8.1 SOFTWARE

The PILOT interpreter is readily adaptable to microprocessor systems. The modularity of the code make enhancement relatively straightforward. The simplified start up procedure linked with computer-controlled lesson loading has taken microprocessor resident PILOT from an interesting technical capability to a useful teaching device, even for uninitiated users.

Changes to software at the machine language level are cumbersome; but if care is taken, useable products can result. Machine language disassemblers greatly aid documentation of machine language coding. Search utilities for locating patterned data rapidly aid in machine language level modifications.

The PHIMON operating system of Digital Group has proven very easy to use and interface. It is well documented and a complete commented assembly language listing of the entire system is provided by Digital Group. Conversely, MAXI BASIC and the Text Editor have poor documentation and no listing of any kind. The inside operations of these two programs were understood only after many hours of disection.

8.2 HARDWARE

The performance of the digital cassette is excellent in this CAI application. The availability of this low-cost computer-controlled mass storage device provides an opportunity to construct very capable CAI systems at minimal cost.

The ROM bootstrap feature of the Digital Group microprocessor makes the system extraordinarily easy to place into operation. This feature is a major capability that optimizes this hardware for CAI operations by inexperienced users. In comparison, the Digital Group System is far easier to begin using than a typical time sharing terminal.

8.3 POTENTIAL ENHANCEMENTS

The following paragraphs present proposals for enhancements that are beyond the scope of the project but appear to be both feasible and desirable. Highest priority for enhancement should be the improvement of the editor's interface with PHIMON for files operations. This enhancement should permit saving an edit file by entering:

"SAVE FILENAME"

or loading an edit file by entering:

"LOAD FILENAME"

where filename is any legal PHIMON filename.

The next priority enhancement should be the addition of a hard copy routine to the PILOT student system. This would permit the author to provide documents for the student's retention. It would be an excellent end point of a lesson to provide the student a hard copy summary of the material covered. This feature will require 300H bytes of additional memory.

Systems with a larger memory capability could have the lesson loading/changing process speeded up greatly by having the PHIMON "Directory" overlay and files directory remain in memory at all times. This would require an additional 400H bytes for the directory.

Other possible enhancements include the incorporation of a BASIC expression analyzer into PILOT to expand the compute statement capability. No research has been conducted on this enhancement. The last priority enhancement might be the incorporation of a PILOT resident editor (such an editor has recently been published [6]). Implementation would require 275H bytes of memory. This line editor would speed up changes to lessons but is far less sophisticated than the editor included in this implementation.

8.4 CONCLUSION

This project has produced a useable system that provides the university with a significant CAI capability. There are currently at least ten Digital Group Systems capable of executing this software on the Manhattan campus. Only the creation of courseware is necessary to realize the potential of this capability. It is the hope of this author that other members of the university will continue this effort. Achievement with this system does not require any depth of knowledge of computers, but requires only the desire to teach and the willingness to try a new media.

APPENDIX A

PILOT LANGUAGE DEFINITION

The following is a Bacus Naur Form definition of Z-80 PILOT as implemented in the project described by this report. The exclamation mark, "!", is used to denote an "OR" relation. Examples of this PILOT language are contained in Appendix B, User Courseware and Appendix G, PILOT Author System Users Manual.

<PILOT program>::=<PILOT line>$_{1}^{N}$

<PILOT line>::=<label>![<label>]<PILOT statement>

<label>::=*<identifier>"carriage-return"!"space"

<IDENTIFIER>::=(<LETTER!DIGIT>)$_{1}^{10}$

<PILOT statement>::=<opcode>[<cond>]:<expression>

<opcode>::=<core-opcode>!<keyword>!<continuation>

<core-opcode>::=T!A!M!J!V!E!C!R!V!N

<key-word>::=MC!INMAX!NEW$!DP!SAVE!
PRINT!EDIT!BYE!
LOAD(<continuation>"space"<filename>)!
(#1 "space" <filename>) NOTE 1

<continuation>::= :

<cond>::=Y!N!<num-variable>!<integer>

<expression>::=<message>!<variable>!<pattern>!
<label>!<assign statement>

<message::=[<text>][<variable>][<text>]

<variable>::=<string variable>!<num-variable>

<pattern::=<text>

<assign-statement>::=<num-variable>=<num-variable>
(+!-)(<integer>!<num-variable>)

<string-variable>::=$<identifier>

<num-variable>::=#<letter>

<integer>::=+!-(<digit>)$_{1}^{2}$

<TEXT>::=(<DIGIT>!<LETTER>!<SPEC-CHAR>)$_{1}^{80}$

<filename>::=<letter>[<letter>!<digit>]$_{0}^{5}$

<digit>::=0!1!2! 9

<letter>::=A!B!C! .... Z

<spec-char>::="!#!$!%!&!´!=!>!<!@!+!-!*!?!;!
.!^!:!left-paren "("!right-paren
")"!"space"!exclamation mark "!"

NOTE 1:

SAVE, PRINT, EDIT are not implemented in this system. They may occur in a PILOT program but the operation will not take place. They are included in the operations recognition logic to assist in transport of programs from other implementations. If they were not in the logic, these key-words would have to be removed from PILOT programs or else the system would have an error condition.

```
R:                         APPENDIX B
R:                      USER COURSEWARE
R:
R:
R:
R:      THE PURPOSE OF THIS APPENDIX IS TO DEMONSTRATE TO PILOT
R:USERS , THE FUNCTION OF THE PILOT SYSTEM . THIS APPENDIX IS
R:PREPARED IN A FORMAT THAT CAN BE EXECUTED ON THE PILOT SYSTEM .
R:CODE CONTAINED IN THIS APPENDIX IS A MODIFIED VERSION OF
R:MATERIAL CONTAINED IN REFERENCE 2.
R:
R:
T:
:
T:                  INTRODUCTION TO PILOT
T:
T:
*BEGIN
T:
T: YOU CAN USE PILOT INSTRUCTIONS:
: T: TO PRESENT TEXT.
: A: TO ACCEPT AN ANSWER.
: M: TO MATCH ELEMENTS OF AN ANSWER.
: J: TO JUMP TO A LABELED PLACE.
: U: TO USE A SUBROUTINE.
: E: TO END A SUBROUTINE OR THE ENTIRE PROGRAM.
: C: FOR LIMITED COMPUTATION.
: R: TO INSERT A REMARK WITH NO OPERATIONAL EFFECT.
: BEGIN BY TYPING ANYTHING, THEN PRESS 'RETURN'
:  USE DELETE   OR RUBOUT TO ERASE ONE LETTER , AND
:  CTL/U TO KILL A LINE BEFORE YOU HAVE PRESSED RETURN.
A:$WHAT
T:YOU TYPED $WHAT.
T:$WHAT IS WHAT YOU TYPED.
T:   PRESS 'RETURN'  TO CONTINUE.
A:
T:
:Y OR N APPENDED TO ANY INSTRUCTION MAKES ITS USE CONDITIONAL
:  UPON THE LAST ATTEMPTED MATCH.
:Y: OR N: ALONE ARE SHORTHAND VERSIONS OF TY: OR TN:
T:
*TESTM
T: I WILL LOOK FOR 'ABC', ' DEF', 'GHI ' OR ' JKL '
T: NOTE THE SPACES. THEY ARE IMPORTANT IF THE LETTERS WE ARE
T:TRYING TO MATCH ARE NOT AT EITHER END OF A LINE.
A:
M:ABC, DEF,GHI , JKL ,
TY:MATCH
TN:NO MATCH
   U:*ASK
   JY:*TESTM
   J:*TESTC


*ASK
T:AGAIN ? (Y OR N)
A:
M: Y
E:
```

```
*TESTC
T:TYPE A NUMBER FROM 0 TO 99
T: I USE THE FIRST TWO DIGITS ENTERED.
   A:#A
T: ANOTHER NUMBER PLEASE
   A:#B
T: YOUR NUMBERS ARE #A AND #B.
T: I WILL SUBTRACT THEM.
    C:C=A-B
T:#A - #B = #C
T:
T:    TYPE 'NEXT' TO GO ON
A:
M:NEXT
 JN:TESTC

T:
*TESTPLUS
  C:A=0
  C:B=0
T:TESTS CAN BE CONDITIONAL ON A NUMERIC VALUE > 0
T:A NUMERIC VARIABLE IS APPENDED AS IN ' J(X):*LABEL'
  T:TYPE 'ABC' OR 'DEF'
A:
M:ABC
 CY:A=1

M:DEF
 CY:B=1


T(A):ABC FOUND
T(B):DEF FOUND
   C:C=A+B
   J(C):*END
T:NO MATCH OF EITHER ABC OR DEF


*END
T:END OF TEST
   U:*ASK
   JY:*TESTPLUS
T:
:FIVE OTHER INSTRUCTIONS ARE AVAILABLE IN THIS SYSTEM
:   MC:      TO LOOK FOR TEXT WITH COMMAS IN IT.
:   INMAX:   TO LIMIT THE NUMBER OF CHARACTERS ACCEPTED.
:   NEW$     TO ERASE STORED $TEXT (STRING VARIABLES).
:   LOAD:    TO LOAD AND RUN A NEW PROGRAM.
:   BYE:     TO LEAVE PILOT AND GO TO A MONITOR.
:
:
:PRESS RETURN TO CONTINUE .
A:
T: WITH THE 'MC' INSTRUCTION ,
   T:YOU CAN LOOK FOR TEXT WITH COMMAS IN IT .
   T:A CARET '+' IS USED AS A SEPARATOR FOR MC:
```

```
*TESTMC
T:I WILL LOOK FOR 'ABC,D' OR 'A,BC'
A:$ENTRY
MC:ABC,D+A,BC
Y:YES,$ENTRY IS OK.
N:NO , YOU TYPED $ENTRY INSTEAD OF THE ABOVE.
     U:*ASK
JY:TESTMC
T:
*TESTIN
T:INMAX:1  WILL BE USED TO LOOK FOR 'A' .
T: PRESS ONLY A SINGLE KEY. DO NOT PRESS 'RETURN'
T:       ----                    ---
INMAX:1
    A:
    M:A
Y: YOU TYPED 'A'
N: THAT WASN'T 'A'
T: AGAIN ?(Y OR N )
A:
INMAX:72
N:Y
 JY:*TESTIN
T:
*ERRORS
T: I WILL SHOW SOME ERROR MESSAGES . AFTER EACH , PRESS 'RETURN'
: I WILL TRY TO JUMP TO AN UNKNOWN LABEL '*HERE '
: PRESS 'RETURN'
A:
 J: *HERE
T:
:IN THE NEXT EXAMPLE, A  C: INSTRUCTION HAS BAD SYNTAX .
T:   PRESS 'RETURN'
A:
    C:X NOT Y
T: THE EXPRESSION FIELD IS DISPLAYED + ERROR MESSAGE.
: PRESS RETURN
A:
T:
: THE VALUE OF X WILL BECOME GREATER THAN 99.
: PRESS RETURN
A:
  C:X=99
C: X=X+10
T: THE STATEMENT CAUSING OVERFLOW IS DISPLAYED.
: THE VALUE X WAS LEFT AT #X
:   A SIMILAR MESSAGE APPEARS WHEN X BECOME SMALLER THAN -99.
T: PRESS RETURN
A:
```

```
T: TYPE 'R' TO REPEAT ERROR MESSAGES.
 INMAX:1
A:
INMAX:72
M:R
 JY:*ERRORS

T:
: AN ILLEGAL STATEMENT IS SIMPLY DISPLAYED.
   MW:*
 T:  'MW ' IS NOT LEGAL IN PILOT.
:
:THAT CONCLUDES THE TESTS.
T: PRESS RETURN TO CONTINUE
: OR ENTER 'NO MORE' TO STOP !
A:
M:NO , MORE,STOP, ENOUGH
JY:*THEEND
J:*BEGIN
*THEEND
BYE:
```

APPENDIX C
DISASSEMBLY LISTING OF Z-80 PILOT

The following pages are a symbolic listing of the Z-80 PILOT interpreter (less the file interface). This listing was produced by the Digital Group Z-80 Disassembler, running in the hex, symbolic mode. The instruction set is in the Zilog Z-80 assembler format. The relative addresses are translated into absolute addresses and placed to the right of the instruction with an asterisk before and after. It is important to note that the disassembler makes no differentiation between valid instructions and data storage. This will typically produce invalid symbolic translation when data storage areas are encountered. For example, the area from 0118H to 02E0H is a system <u>data</u> area (see Figure 6.1). The disassembler translated this area as if it contained valid instructions (see pages C-6 through C-13).

```
0000 C3 00 E0 JP    E000
0003 00          NOP
0004 00          NOP
0005 00          NOP
0006 00          NOP
0007 00          NOP
0008 00          NOP
0009 00          NOP
000A 00          NOP
000B 00          NOP
000C 00          NOP
000D 00          NOP
000E 00          NOP
000F 00          NOP
0010 00          NOP
0011 00          NOP
0012 00          NOP
0013 00          NOP
0014 00          NOP
0015 00          NOP
0016 00          NOP
0017 00          NOP
0018 00          NOP
0019 00          NOP
001A 00          NOP
001B 00          NOP
001C 00          NOP
001D 00          NOP
001E 00          NOP
001F 00          NOP
0020 00          NOP
0021 00          NOP
0022 00          NOP
0023 00          NOP
0024 00          NOP
0025 00          NOP
0026 00          NOP
0027 00          NOP
0028 00          NOP
0029 00          NOP
002A 00          NOP
002B 00          NOP
002C 00          NOP
002D 00          NOP
002E 00          NOP
002F 00          NOP
0030 C3 13 E0 JP    E013
0033 00          NOP
0034 00          NOP
0035 00          NOP
0036 00          NOP
0037 00          NOP
0038 00          NOP
0039 00          NOP
003A 00          NOP
003B 00          NOP
003C 00          NOP
003D 00          NOP
003E 00          NOP
```

```
003F 00          NOP
0040 00          NOP
0041 00          NOP
0042 00          NOP
0043 00          NOP
0044 00          NOP
0045 00          NOP
0046 00          NOP
0047 00          NOP
0048 00          NOP
0049 00          NOP
004A 00          NOP
004B 00          NOP
004C 00          NOP
004D 00          NOP
004E 00          NOP
004F 00          NOP
0050 00          NOP
0051 00          NOP
0052 00          NOP
0053 00          NOP
0054 00          NOP
0055 00          NOP
0056 00          NOP
0057 00          NOP
0058 00          NOP
0059 00          NOP
005A 00          NOP
005B 00          NOP
005C 00          NOP
005D 00          NOP
005E 00          NOP
005F 00          NOP
0060 00          NOP
0061 00          NOP
0062 00          NOP
0063 00          NOP
0064 00          NOP
0065 00          NOP
0066 00          NOP
0067 00          NOP
0068 00          NOP
0069 00          NOP
006A 00          NOP
006B 00          NOP
006C 00          NOP
006D 00          NOP
006E 00          NOP
006F 00          NOP
0070 00          NOP
0071 00          NOP
0072 00          NOP
0073 00          NOP
0074 00          NOP
0075 00          NOP
0076 00          NOP
0077 00          NOP
0078 00          NOP
```

```
0078 00          NOP
0079 00          NOP
007A 00          NOP
007B 00          NOP
007C 00          NOP
007D 00          NOP
007E 00          NOP
007F 00          NOP
0080 00          NOP
0081 00          NOP
0082 00          NOP
0083 00          NOP
0084 00          NOP
0085 00          NOP
0086 00          NOP
0087 00          NOP
0088 00          NOP
0089 00          NOP
008A 00          NOP
008B 00          NOP
008C 00          NOP
008D 00          NOP
008E 00          NOP
008F 00          NOP
0090 00          NOP
0091 00          NOP
0092 00          NOP
0093 00          NOP
0094 00          NOP
0095 00          NOP
0096 00          NOP
0097 00          NOP
0098 00          NOP
0099 00          NOP
009A 00          NOP
009B 00          NOP
009C 00          NOP
009D 00          NOP
009E 00          NOP
009F 00          NOP
00A0 00          NOP
00A1 00          NOP
00A2 00          NOP
00A3 00          NOP
00A4 00          NOP
00A5 00          NOP
00A6 00          NOP
00A7 00          NOP
00A8 00          NOP
00A9 00          NOP
00AA 00          NOP
00AB 00          NOP
00AC 00          NOP
00AD 00          NOP
00AE 00          NOP
00AF 00          NOP
00B0 00          NOP
00B1 00          NOP
00B2 00          NOP
```

```
00B3 00       NOP
00B4 00       NOP
00B5 00       NOP
00B6 00       NOP
00B7 00       NOP
00B8 00       NOP
00B9 00       NOP
00BA 00       NOP
00BB 00       NOP
00BC 00       NOP
00BD 00       NOP
00BE 00       NOP
00BF 00       NOP
00C0 00       NOP
00C1 00       NOP
00C2 00       NOP
00C3 00       NOP
00C4 00       NOP
00C5 00       NOP
00C6 00       NOP
00C7 00       NOP
00C8 00       NOP
00C9 00       NOP
00CA 00       NOP
00CB 00       NOP
00CC 00       NOP
00CD 00       NOP
00CE 00       NOP
00CF 00       NOP
00D0 00       NOP
00D1 00       NOP
00D2 00       NOP
00D3 00       NOP
00D4 00       NOP
00D5 00       NOP
00D6 00       NOP
00D7 00       NOP
00D8 00       NOP
00D9 00       NOP
00DA 00       NOP
00DB 00       NOP
00DC A0       AND  B
00DD 0F       RRCA
00DE 87       ADD  A
00DF 0F       RRCA
00E0 85       ADD  L
00E1 E3       EX   (SP),(HL)
00E2 85       ADD  L
00E3 E3       EX   (SP),(HL)
00E4 A0       AND  B
00E5 E3       EX   (SP),(HL)
00E6 91       SUB  C
00E7 E3       EX   (SP),(HL)
00E8 22 E0 73 LD   (73E0),HL
00EB E5       PUSH HL
00EC 60       LD   H,B
00ED 00       NOP
```

```
00EE CF          RST  08
00EF E5          PUSH HL
00F0 00          NOP
00F1 0A          LD   A,(BC)
00F2 6F          LD   L,A
00F3 E1          POP  HL
00F4 08          EX   AF,AF'
00F5 14          INC  D
00F6 EA 0B FA    JP   PE,FA0B
00F9 04          INC  B
00FA F9          LD   SP,HL
00FB 03          INC  BC
00FC B7          OR   A
00FD 03          INC  BC
00FE 18 03       JR   03        *0103*
0100 C3 A3 0F    JP   0FA3
0103 C3 A4 0D    JP   0DA4
0106 C3 A3 0F    JP   0FA3
0109 C3 A4 0D    JP   0DA4
010C C3 A4 0D    JP   0DA4
010F C3 60 E0    JP   E060
0112 C3 60 E0    JP   E060
0115 C3 D9 0B    JP   0BD9
0118 0B          DEC  BC
0119 15          DEC  D
011A 2B          DEC  HL
011B 15          DEC  D
011C EE 01       XOR  01
011E 00          NOP
011F 00          NOP
0120 4C          LD   C,H
0121 01 25 15    LD   BC,1525
0124 00          NOP
0125 00          NOP
0126 1B          DEC  DE
0127 4C          LD   C,H
0128 00          NOP
0129 00          NOP
012A 00          NOP
012B 00          NOP
012C 00          NOP
012D 00          NOP
012E 00          NOP
012F 00          NOP
0130 00          NOP
0131 00          NOP
0132 00          NOP
0133 00          NOP
0134 FE 3B       CP   3B
0136 A1          AND  C
0137 01 1C 15    LD   BC,151C
013A 30 15       JR   NC,15        *0151*
013C 16 19       LD   D,19
013E FF          RST  56
013F 3B          DEC  SP
0140 03          INC  BC
0141 01 00 15    LD   BC,1500
0144 48          LD   C,B
```

```
0145 00       NOP
0146 00       NOP
0147 2C       INC  L
0148 03       INC  BC
0149 00       NOP
014A 42       LD   B,D
014B 00       NOP
014C 46       LD   B,(HL)
014D 49       LD   C,C
014E 4C       LD   C,H
014F 45       LD   B,L
0150 20 4F    JR   NZ,4F        *01A1*
0152 4E       LD   C,(HL)
0153 45       LD   B,L
0154 0D       DEC  C
0155 0D       DEC  C
0156 4C       LD   C,H
0157 4F       LD   C,A
0158 54       LD   D,H
0159 20 43    JR   NZ,43        *019E*
015B 48       LD   C,B
015C 41       LD   B,C
015D 49       LD   C,C
015E 4E       LD   C,(HL)
015F 49       LD   C,C
0160 4E       LD   C,(HL)
0161 47       LD   B,A
0162 0D       DEC  C
0163 0D       DEC  C
0164 CE B1    ADC  B1
0166 00       NOP
0167 49       LD   C,C
0168 45       LD   B,L
0169 50       LD   D,B
016A 3A 0D 00 LD   A,(0D0D)
016D 52       LD   D,D
016E 20 46    JR   NZ,46        *01B6*
0170 4F       LD   C,A
0171 52       LD   D,D
0172 20 4D    JR   NZ,4D        *01C1*
0174 43       LD   B,E
0175 3A 0D 0D LD   A,(0D0D)
0178 52       LD   D,D
0179 2E 0D    LD   L,0D
017B 0D       DEC  C
017C 43       LD   B,E
017D 45       LD   B,L
017E 2E 0D    LD   L,0D
0180 0D       DEC  C
0181 0D       DEC  C
0182 0D       DEC  C
0183 0D       DEC  C
0184 52       LD   D,D
0185 45       LD   B,L
0186 0D       DEC  C
0187 0D       DEC  C
0188 0D       DEC  C
```

```
0189 0D        DEC  C
018A 00        NOP
018B 00        NOP
018C 00        NOP
018D 00        NOP
018E 00        NOP
018F 00        NOP
0190 00        NOP
0191 00        NOP
0192 00        NOP
0193 00        NOP
0194 00        NOP
0195 00        NOP
0196 00        NOP
0197 00        NOP
0198 00        NOP
0199 00        NOP
019A 00        NOP
019B 00        NOP
019C 00        NOP
019D 20 4E     JR   NZ,4E       *01ED*
019F 20 0D     JR   NZ,0D       *01AE*
01A1 20 0D     JR   NZ,0D       *01B0*
01A3 0D        DEC  C
01A4 0D        DEC  C
01A5 00        NOP
01A6 00        NOP
01A7 00        NOP
01A8 00        NOP
01A9 00        NOP
01AA 00        NOP
01AB 00        NOP
01AC 00        NOP
01AD 00        NOP
01AE 00        NOP
01AF 00        NOP
01B0 00        NOP
01B1 00        NOP
01B2 00        NOP
01B3 00        NOP
01B4 00        NOP
01B5 00        NOP
01B6 00        NOP
01B7 00        NOP
01B8 00        NOP
01B9 00        NOP
01BA 00        NOP
01BB 00        NOP
01BC 00        NOP
01BD 00        NOP
01BE 00        NOP
01BF 00        NOP
01C0 00        NOP
01C1 00        NOP
01C2 00        NOP
01C3 00        NOP
01C4 00        NOP
01C5 00        NOP
```

```
01C6 00         NOP
01C7 00         NOP
01C8 00         NOP
01C9 00         NOP
01CA 00         NOP
01CB 00         NOP
01CC 00         NOP
01CD 00         NOP
01CE 00         NOP
01CF 00         NOP
01D0 00         NOP
01D1 00         NOP
01D2 00         NOP
01D3 00         NOP
01D4 00         NOP
01D5 00         NOP
01D6 00         NOP
01D7 00         NOP
01D8 00         NOP
01D9 00         NOP
01DA 00         NOP
01DB 00         NOP
01DC 00         NOP
01DD 00         NOP
01DE 00         NOP
01DF 00         NOP
01E0 00         NOP
01E1 00         NOP
01E2 00         NOP
01E3 00         NOP
01E4 00         NOP
01E5 00         NOP
01E6 00         NOP
01E7 00         NOP
01E8 00         NOP
01E9 00         NOP
01EA 00         NOP
01EB 00         NOP
01EC 00         NOP
01ED 00         NOP
01EE 20 59      JR   NZ,59       *0249*
01F0 0D         DEC  C
01F1 0D         DEC  C
01F2 0D         DEC  C
01F3 00         NOP
01F4 00         NOP
01F5 00         NOP
01F6 00         NOP
01F7 00         NOP
01F8 00         NOP
01F9 00         NOP
01FA 00         NOP
01FB 00         NOP
01FC 00         NOP
01FD 00         NOP
01FE 00         NOP
01FF 00         NOP
0200 00         NOP
```

```
0201 00          NOP
0202 00          NOP
0203 00          NOP
0204 00          NOP
0205 00          NOP
0206 00          NOP
0207 00          NOP
0208 00          NOP
0209 00          NOP
020A 00          NOP
020B 00          NOP
020C 00          NOP
020D 00          NOP
020E 00          NOP
020F 00          NOP
0210 00          NOP
0211 00          NOP
0212 00          NOP
0213 00          NOP
0214 00          NOP
0215 00          NOP
0216 00          NOP
0217 00          NOP
0218 00          NOP
0219 00          NOP
021A 00          NOP
021B 00          NOP
021C 00          NOP
021D 00          NOP
021E 00          NOP
021F 00          NOP
0220 00          NOP
0221 00          NOP
0222 00          NOP
0223 00          NOP
0224 00          NOP
0225 00          NOP
0226 00          NOP
0227 00          NOP
0228 00          NOP
0229 00          NOP
022A 00          NOP
022B 00          NOP
022C 00          NOP
022D 00          NOP
022E 00          NOP
022F 00          NOP
0230 00          NOP
0231 00          NOP
0232 00          NOP
0233 00          NOP
0234 00          NOP
0235 00          NOP
0236 00          NOP
0237 00          NOP
0238 00          NOP
0239 00          NOP
023A 00          NOP
```

```
023B 00         NOP
023C 00         NOP
023D 00         NOP
023E 00         NOP
023F 24         INC   H
0240 57         LD    D,A
0241 48         LD    C,B
0242 41         LD    B,C
0243 54         LD    D,H
0244 0D         DEC   C
0245 0D         DEC   C
0246 55         LD    D,L
0247 53         LD    D,E
0248 0D         DEC   C
0249 00         NOP
024A 00         NOP
024B 4C         LD    C,H
024C 4F         LD    C,A
024D 41         LD    B,C
024E 44         LD    B,H
024F 20 3A      JR    NZ,3A        *028B*
0251 00         NOP
0252 00         NOP
0253 00         NOP
0254 00         NOP
0255 00         NOP
0256 41         LD    B,C
0257 00         NOP
0258 42         LD    B,D
0259 00         NOP
025A 43         LD    B,E
025B 00         NOP
025C 44         LD    B,H
025D 00         NOP
025E 45         LD    B,L
025F 00         NOP
0260 46         LD    B,(HL)
0261 00         NOP
0262 47         LD    B,A
0263 00         NOP
0264 48         LD    C,B
0265 00         NOP
0266 49         LD    C,C
0267 00         NOP
0268 4A         LD    C,D
0269 00         NOP
026A 4B         LD    C,E
026B 00         NOP
026C 4C         LD    C,H
026D 00         NOP
026E 4D         LD    C,L
026F 00         NOP
0270 4E         LD    C,(HL)
0271 00         NOP
0272 4F         LD    C,A
0273 00         NOP
0274 50         LD    D,B
0275 00         NOP
```

```
0276 51          LD   D,C
0277 00          NOP
0278 52          LD   D,D
0279 00          NOP
027A 53          LD   D,E
027B 00          NOP
027C 54          LD   D,H
027D 00          NOP
027E 55          LD   D,L
027F 00          NOP
0280 56          LD   D,(HL)
0281 00          NOP
0282 57          LD   D,A
0283 00          NOP
0284 58          LD   E,B
0285 00          NOP
0286 59          LD   E,C
0287 00          NOP
0288 5A          LD   E,D
0289 00          NOP
028A 01 4C 4F    LD   BC,4F4C
028D 41          LD   B,C
028E 44          LD   B,H
028F 0D          DEC  C
0290 0D          DEC  C
0291 20 0D       JR   NZ,0D        *02A0*
0293 53          LD   D,E
0294 0D          DEC  C
0295 00          NOP
0296 00          NOP
0297 00          NOP
0298 00          NOP
0299 00          NOP
029A 00          NOP
029B 00          NOP
029C 00          NOP
029D 00          NOP
029E 00          NOP
029F 00          NOP
02A0 00          NOP
02A1 00          NOP
02A2 00          NOP
02A3 00          NOP
02A4 00          NOP
02A5 00          NOP
02A6 00          NOP
02A7 00          NOP
02A8 00          NOP
02A9 00          NOP
02AA 00          NOP
02AB 00          NOP
02AC 00          NOP
02AD 00          NOP
02AE 00          NOP
02AF 00          NOP
02B0 00          NOP
02B1 00          NOP
02B2 00          NOP
```

```
02B3 00       NOP
02B4 00       NOP
02B5 00       NOP
02B6 00       NOP
02B7 00       NOP
02B8 00       NOP
02B9 00       NOP
02BA 00       NOP
02BB 00       NOP
02BC 00       NOP
02BD 00       NOP
02BE 00       NOP
02BF 00       NOP
02C0 00       NOP
02C1 00       NOP
02C2 00       NOP
02C3 00       NOP
02C4 00       NOP
02C5 00       NOP
02C6 00       NOP
02C7 00       NOP
02C8 00       NOP
02C9 00       NOP
02CA 00       NOP
02CB 00       NOP
02CC 00       NOP
02CD 00       NOP
02CE 00       NOP
02CF 00       NOP
02D0 00       NOP
02D1 00       NOP
02D2 00       NOP
02D3 00       NOP
02D4 00       NOP
02D5 00       NOP
02D6 00       NOP
02D7 00       NOP
02D8 00       NOP
02D9 00       NOP
02DA 00       NOP
02DB 00       NOP
02DC 00       NOP
02DD 00       NOP
02DE 00       NOP
02DF 00       NOP
02E0 31 00 01 LD   SP,0100
02E3 21 FE 3B LD   HL,3BFE
02E6 22 34 01 LD   (0134),HL
02E9 21 F7 02 LD   HL,02F7
02EC 11 00 01 LD   DE,0100
02EF 0E 18    LD   C,18
02F1 CD 54 0C CALL 0C54
02F4 C3 0F 03 JP   030F
02F7 C3 A3 0F JP   0FA3
02FA C3 A4 0D JP   0DA4
02FD C3 A3 0F JP   0FA3
0300 C3 A4 0D JP   0DA4
0303 C3 A4 0D JP   0DA4
```

```
0306 C3 60 E0 JP   E060
0309 C3 60 E0 JP   E060
030C C3 D9 0B JP   0BD9
030F 31 00 01 LD   SP,0100
0312 CD 56 03 CALL 0356
0315 CD 80 03 CALL 0380
0318 C3 0F 03 JP   030F
031B 00       NOP
031C 00       NOP
031D 00       NOP
031E 00       NOP
031F 00       NOP
0320 00       NOP
0321 00       NOP
0322 00       NOP
0323 00       NOP
0324 00       NOP
0325 00       NOP
0326 00       NOP
0327 00       NOP
0328 00       NOP
0329 00       NOP
032A 00       NOP
032B 00       NOP
032C 00       NOP
032D 00       NOP
032E 00       NOP
032F 00       NOP
0330 00       NOP
0331 00       NOP
0332 00       NOP
0333 00       NOP
0334 00       NOP
0335 00       NOP
0336 00       NOP
0337 00       NOP
0338 00       NOP
0339 00       NOP
033A 00       NOP
033B 00       NOP
033C 00       NOP
033D 00       NOP
033E 00       NOP
033F 00       NOP
0340 00       NOP
0341 00       NOP
0342 00       NOP
0343 00       NOP
0344 00       NOP
0345 00       NOP
0346 00       NOP
0347 00       NOP
0348 00       NOP
0349 00       NOP
034A 00       NOP
034B 00       NOP
034C 00       NOP
034D 00       NOP
```

```
034E 00          NOP
034F 00          NOP
0350 00          NOP
0351 00          NOP
0352 48          LD   C,B
0353 45          LD   B,L
0354 52          LD   D,D
0355 0D          DEC  C
0356 21 00 15 LD   HL,1500
0359 22 3A 01 LD   (013A),HL
035C 22 42 01 LD   (0142),HL
035F 21 FF 3B LD   HL,3BFF
0362 22 3E 01 LD   (013E),HL
0365 36 01       LD   (HL),01
0367 C9          RET
0368 CD FD 0B CALL 0BFD
036B 36 01       LD   (HL),01
036D 3E 48       LD   A,48
036F 32 44 01 LD   (0144),A
0372 AF          XOR  A
0373 32 45 01 LD   (0145),A
0376 32 46 01 LD   (0146),A
0379 21 03 01 LD   HL,0103
037C 22 40 01 LD   (0140),HL
037F C9          RET
0380 2A 3A 01 LD   HL,(013A)
0383 7E          LD   A,(HL)
0384 FE 01       CP   01
0386 C8          RET  Z
0387 FE 0D       CP   0D
0389 C2 90 03 JP   NZ,0390
038C 23          INC  HL
038D C3 83 03 JP   0383
0390 CD 9D 07 CALL 079D
0393 FE 01       CP   01
0395 C8          RET  Z
0396 23          INC  HL
0397 22 3A 01 LD   (013A),HL
039A 2B          DEC  HL
039B CD AE 07 CALL 07AE
039E CD 7A 04 CALL 047A
03A1 CD 2C 07 CALL 072C
03A4 FE 3A       CP   3A
03A6 CC BA 03 CALL Z,03BA
03A9 CA 80 03 JP   Z,0380
03AC FE 2A       CP   2A
03AE CC 3C 07 CALL Z,073C
03B1 CA 83 03 JP   Z,0383
03B4 CD CE 03 CALL 03CE
03B7 C3 80 03 JP   0380
03BA 23          INC  HL
03BB 22 1A 01 LD   (011A),HL
03BE 21 4B 02 LD   HL,024B
03C1 22 22 01 LD   (0122),HL
03C4 06 3A       LD   B,3A
03C6 CD 37 08 CALL 0837
03C9 CD E6 03 CALL 03E6
03CC AF          XOR  A
```

```
03CD C9          RET
03CE 22 22 01 LD    (0122),HL
03D1 CD 8E 07 CALL 078E
03D4 2A 22 01 LD    HL,(0122)
03D7 06 3A    LD    B,3A
03D9 CD 37 08 CALL 0837
03DC 79       LD    A,C
03DD B7       OR    A
03DE CA FC 03 JP    Z,03FC
03E1 23       INC   HL
03E2 22 1A 01 LD    (011A),HL
03E5 2B       DEC   HL
03E6 2B       DEC   HL
03E7 CD 84 04 CALL 0484
03EA B7       OR    A
03EB C8       RET   Z
03EC CD A7 04 CALL 04A7
03EF C8       RET   Z
03F0 2A 22 01 LD    HL,(0122)
03F3 CD 71 07 CALL 0771
03F6 CD E6 04 CALL 04E6
03F9 FE 01    CP    01
03FB C0       RET   NZ
03FC 2A 22 01 LD    HL,(0122)
03FF CD 15 01 CALL 0115
0402 C8       RET   Z
0403 2A 22 01 LD    HL,(0122)
0406 22 1A 01 LD    (011A),HL
0409 CD 79 08 CALL 0879
040C C9       RET
040D 21 00 15 LD    HL,1500
0410 CD 9D 07 CALL 079D
0413 23       INC   HL
0414 7E       LD    A,(HL)
0415 FE 0A    CP    0A
0417 C2 2C 04 JP    NZ,042C
041A CD 9D 07 CALL 079D
041D 23       INC   HL
041E 7E       LD    A,(HL)
041F FE 0A    CP    0A
0421 C2 2C 04 JP    NZ,042C
0424 3A 46 01 LD    A,(0146)
0427 C6 01    ADD   01
0429 32 46 01 LD    (0146),A
042C 21 00 15 LD    HL,1500
042F CD 9D 07 CALL 079D
0432 23       INC   HL
0433 CD 7A 04 CALL 047A
0436 5E       LD    E,(HL)
0437 16 00    LD    D,00
0439 1B       DEC   DE
043A 19       ADD   HL,DE
043B 7E       LD    A,(HL)
043C FE 0D    CP    0D
043E C2 56 04 JP    NZ,0456
0441 23       INC   HL
0442 CD 7A 04 CALL 047A
0445 5E       LD    E,(HL)
```

```
0446 1B          DEC  DE
0447 19          ADD  HL,DE
0448 7E          LD   A,(HL)
0449 FE 0D       CP   0D
044B C2 56 04 JP   NZ,0456
044E 3A 46 01 LD   A,(0146)
0451 C6 01       ADD  01
0453 32 46 01 LD   (0146),A
0456 21 01 15 LD   HL,1501
0459 CD 9D 07 CALL 079D
045C 23          INC  HL
045D CD 7A 04 CALL 047A
0460 7E          LD   A,(HL)
0461 CD 6E 04 CALL 046E
0464 C0          RET  NZ
0465 3A 46 01 LD   A,(0146)
0468 C6 04       ADD  04
046A 32 46 01 LD   (0146),A
046D C9          RET
046E FE 30       CP   30
0470 F8          RET  M
0471 FE 3A       CP   3A
0473 FA 78 04 JP   M,0478
0476 B4          OR   H
0477 C9          RET
0478 AF          XOR  A
0479 C9          RET
047A 3A 46 01 LD   A,(0146)
047D B7          OR   A
047E C8          RET  Z
047F 23          INC  HL
0480 3D          DEC  A
0481 C3 7D 04 JP   047D
0484 CD 34 07 CALL 0734
0487 FE 59       CP   59
0489 CA 93 04 JP   Z,0493
048C FE 4E       CP   4E
048E CA 9C 04 JP   Z,049C
0491 B7          OR   A
0492 C9          RET
0493 3A 4B 01 LD   A,(014B)
0496 B7          OR   A
0497 CA A5 04 JP   Z,04A5
049A B4          OR   H
049B C9          RET
049C 3A 4B 01 LD   A,(014B)
049F B7          OR   A
04A0 C2 A5 04 JP   NZ,04A5
04A3 B4          OR   H
04A4 C9          RET
04A5 AF          XOR  A
04A6 C9          RET
04A7 FE 29       CP   29
04A9 CA AE 04 JP   Z,04AE
04AC B4          OR   H
04AD C9          RET
04AE 2B          DEC  HL
04AF 2B          DEC  HL
```

```
04B0 7E       LD   A,(HL)
04B1 FE 28    CP   28
04B3 C2 CB 04 JP   NZ,04CB
04B6 23       INC  HL
04B7 46       LD   B,(HL)
04B8 CD D8 04 CALL 04D8
04BB FE 01    CP   01
04BD CA CB 04 JP   Z,04CB
04C0 23       INC  HL
04C1 7E       LD   A,(HL)
04C2 FE 01    CP   01
04C4 FA C9 04 JP   M,04C9
04C7 B4       OR   H
04C8 C9       RET
04C9 AF       XOR  A
04CA C9       RET
04CB 2A 22 01 LD   HL,(0122)
04CE CD 7C 08 CALL 087C
04D1 21 08 0E LD   HL,0E08
04D4 CD DE 0D CALL 0DDE
04D7 C9       RET
04D8 21 56 02 LD   HL,0256
04DB 7E       LD   A,(HL)
04DC FE 01    CP   01
04DE C8       RET  Z
04DF B8       CP   B
04E0 C8       RET  Z
04E1 23       INC  HL
04E2 23       INC  HL
04E3 C3 DB 04 JP   04DB
04E6 11 11 05 LD   DE,0511
04E9 CD FC 04 CALL 04FC
04EC FE 01    CP   01
04EE C8       RET  Z
04EF EB       EX   DE,(HL)
04F0 23       INC  HL
04F1 11 FA 04 LD   DE,04FA
04F4 D5       PUSH DE
04F5 5E       LD   E,(HL)
04F6 23       INC  HL
04F7 56       LD   D,(HL)
04F8 D5       PUSH DE
04F9 C9       RET
04FA AF       XOR  A
04FB C9       RET
04FC 21 8B 02 LD   HL,028B
04FF CD FB 06 CALL 06FB
0502 B7       OR   A
0503 C0       RET  NZ
0504 23       INC  HL
0505 EB       EX   DE,(HL)
0506 23       INC  HL
0507 23       INC  HL
0508 23       INC  HL
0509 7E       LD   A,(HL)
050A FE 01    CP   01
050C C8       RET  Z
050D EB       EX   DE,(HL)
```

```
050E C3 FC 04 JP   04FC
0511 54       LD   D,H
0512 0D       DEC  C
0513 79       LD   A,C
0514 08       EX   AF,AF'
0515 41       LD   B,C
0516 0D       DEC  C
0517 58       LD   E,B
0518 09       ADD  HL,BC
0519 4D       LD   C,L
051A 0D       DEC  C
051B 26 0A    LD   H,0A
051D 4D       LD   C,L
051E 43       LD   B,E
051F 0D       DEC  C
0520 1E 0A    LD   E,0A
0522 4A       LD   C,D
0523 0D       DEC  C
0524 CB 05    RLC  L
0526 52       LD   D,D
0527 0D       DEC  C
0528 DB 0B    IN   0B
052A 43       LD   B,E
052B 0D       DEC  C
052C 09       ADD  HL,BC
052D 0B       DEC  BC
052E 55       LD   D,L
052F 0D       DEC  C
0530 08       EX   AF,AF'
0531 06 45    LD   B,45
0533 0D       DEC  C
0534 DC 0B 59 CALL C,590B
0537 0D       DEC  C
0538 79       LD   A,C
0539 08       EX   AF,AF'
053A 4E       LD   C,(HL)
053B 0D       DEC  C
053C 79       LD   A,C
053D 08       EX   AF,AF'
053E 4C       LD   C,H
053F 4F       LD   C,A
0540 41       LD   B,C
0541 44       LD   B,H
0542 0D       DEC  C
0543 E7       RST  32
0544 0B       DEC  BC
0545 49       LD   C,C
0546 4E       LD   C,(HL)
0547 4D       LD   C,L
0548 41       LD   B,C
0549 58       LD   E,B
054A 0D       DEC  C
054B 89       ADC  C
054C 05       DEC  B
054D 4E       LD   C,(HL)
054E 45       LD   B,L
054F 57       LD   D,A
0550 24       INC  H
```

```
0551 0D          DEC  C
0552 EE 0B       XOR  0B
0554 44          LD   B,H
0555 50          LD   D,B
0556 0D          DEC  C
0557 7A          LD   A,D
0558 0D          DEC  C
0559 50          LD   D,B
055A 52          LD   D,D
055B 49          LD   C,C
055C 4E          LD   C,(HL)
055D 54          LD   D,H
055E 0D          DEC  C
055F 84          ADD  H
0560 0D          DEC  C
0561 53          LD   D,E
0562 41          LD   B,C
0563 56          LD   D,(HL)
0564 45          LD   B,L
0565 0D          DEC  C
0566 94          SUB  H
0567 0D          DEC  C
0568 49          LD   C,C
0569 45          LD   B,L
056A 50          LD   D,B
056B 0D          DEC  C
056C 7C          LD   A,H
056D 05          DEC  B
056E 42          LD   B,D
056F 59          LD   E,C
0570 45          LD   B,L
0571 0D          DEC  C
0572 0F          RRCA
0573 01 45 44 LD   BC,4445
0576 49          LD   C,C
0577 54          LD   D,H
0578 0D          DEC  C
0579 12          LD   (DE),A
057A 01 01 21 LD   BC,2101
057D 00          NOP
057E 15          DEC  D
057F 22 3A 01 LD   (013A),HL
0582 22 42 01 LD   (0142),HL
0585 CD 0D 04 CALL 040D
0588 C9          RET
0589 CD 9A 05 CALL 059A
058C 7B          LD   A,E
058D FE 49       CP   49
058F FA 95 05 JP   N,0595
0592 3E 48       LD   A,48
0594 5F          LD   E,A
0595 21 44 01 LD   HL,0144
0598 73          LD   (HL),E
0599 C9          RET
059A 2A 1A 01 LD   HL,(011A)
059D CD 2C 07 CALL 072C
05A0 FE 0D       CP   0D
```

```
05A2 C8       RET  Z
05A3 CD 3C 07 CALL 073C
05A6 2B       DEC  HL
05A7 22 1A 01 LD   (011A),HL
05AA 21 8B 02 LD   HL,028B
05AD CD 60 08 CALL 0860
05B0 C2 C2 05 JP   NZ,05C2
05B3 46       LD   B,(HL)
05B4 CD D8 04 CALL 04D8
05B7 FE 01    CP   01
05B9 CC CB 04 CALL Z,04CB
05BC C8       RET  Z
05BD 23       INC  HL
05BE 5E       LD   E,(HL)
05BF C3 C5 05 JP   05C5
05C2 CD 31 06 CALL 0631
05C5 7B       LD   A,E
05C6 B7       OR   A
05C7 F0       RET  P
05C8 1E 00    LD   E,00
05CA C9       RET
05CB 2A 1A 01 LD   HL,(011A)
05CE CD 2C 07 CALL 072C
05D1 FE 2A    CP   2A
05D3 CA E7 05 JP   Z,05E7
05D6 3E 2A    LD   A,2A
05D8 32 8B 02 LD   (028B),A
05DB 11 8C 02 LD   DE,028C
05DE 2A 1A 01 LD   HL,(011A)
05E1 CD 54 07 CALL 0754
05E4 C3 EA 05 JP   05EA
05E7 CD 3C 07 CALL 073C
05EA 2A 42 01 LD   HL,(0142)
05ED CD 9D 06 CALL 069D
05F0 FE 01    CP   01
05F2 CC FB 05 CALL Z,05FB
05F5 C8       RET  Z
05F6 23       INC  HL
05F7 22 3A 01 LD   (013A),HL
05FA C9       RET
05FB 21 8B 02 LD   HL,028B
05FE CD 49 0D CALL 0D49
0601 21 E3 0D LD   HL,0DE3
0604 CD DE 0D CALL 0DDE
0607 C9       RET
0608 CD 13 0C CALL 0C13
060B C3 CB 05 JP   05CB
060E 70       LD   (HL),B
060F 23       INC  HL
0610 79       LD   A,C
0611 D6 01    SUB  01
0613 4F       LD   C,A
0614 C2 0E 06 JP   NZ,060E
0617 C9       RET
0618 2A 34 01 LD   HL,(0134)
061B EB       EX   DE,(HL)
061C 21 00 15 LD   HL,1500
061F 06 2A    LD   B,2A
```

```
0621 70          LD    (HL),B
0622 23          INC   HL
0623 CD 2A 06    CALL  062A
0626 C2 21 06    JP    NZ,0621
0629 C9          RET
062A 7C          LD    A,H
062B BA          CP    D
062C F8          RET   N
062D C0          RET   NZ
062E 7D          LD    A,L
062F BB          CP    E
0630 C9          RET
0631 1E 00       LD    E,00
0633 23          INC   HL
0634 CD B9 07    CALL  07B9
0637 CA 61 06    JP    Z,0661
063A FE 2B       CP    2B
063C CA 61 06    JP    Z,0661
063F FE 2D       CP    2D
0641 CA 61 06    JP    Z,0661
0644 2B          DEC   HL
0645 7E          LD    A,(HL)
0646 FE 30       CP    30
0648 F8          RET   N
0649 FE 3A       CP    3A
064B F0          RET   P
064C D6 30       SUB   30
064E 87          ADD   A
064F 5F          LD    E,A
0650 87          ADD   A
0651 87          ADD   A
0652 83          ADD   E
0653 5F          LD    E,A
0654 23          INC   HL
0655 7E          LD    A,(HL)
0656 FE 30       CP    30
0658 F8          RET   N
0659 FE 3A       CP    3A
065B F0          RET   P
065C D6 30       SUB   30
065E 83          ADD   E
065F 5F          LD    E,A
0660 C9          RET
0661 2B          DEC   HL
0662 C3 55 06    JP    0655
0665 0E 00       LD    C,00
0667 7B          LD    A,E
0668 B7          OR    A
0669 FC 8E 06    CALL  N,068E
066C FE 0A       CP    0A
066E FA 7B 06    JP    N,067B
0671 D6 0A       SUB   0A
0673 5F          LD    E,A
0674 79          LD    A,C
0675 C6 01       ADD   01
0677 4F          LD    C,A
0678 C3 67 06    JP    0667
067B 79          LD    A,C
```

```
067C C6 30    ADD  30
067E FE 30    CP   30
0680 CA 85 06 JP   Z,0685
0683 77       LD   (HL),A
0684 23       INC  HL
0685 7B       LD   A,E
0686 C6 30    ADD  30
0688 77       LD   (HL),A
0689 23       INC  HL
068A 3E 0D    LD   A,0D
068C 77       LD   (HL),A
068D C9       RET
068E 3E 2D    LD   A,2D
0690 77       LD   (HL),A
0691 23       INC  HL
0692 AF       XOR  A
0693 93       SUB  E
0694 5F       LD   E,A
0695 C9       RET
0696 AF       XOR  A
0697 32 49 01 LD   (0149),A
069A C3 A4 06 JP   06A4
069D B4       OR   H
069E 32 49 01 LD   (0149),A
06A1 CD 7A 04 CALL 047A
06A4 CD 2C 07 CALL 072C
06A7 FE 01    CP   01
06A9 C8       RET  Z
06AA FE 2A    CP   2A
06AC CA C2 06 JP   Z,06C2
06AF FE 24    CP   24
06B1 CA C2 06 JP   Z,06C2
06B4 CD 9D 07 CALL 079D
06B7 23       INC  HL
06B8 3A 49 01 LD   A,(0149)
06BB B7       OR   A
06BC CA A4 06 JP   Z,06A4
06BF C3 A1 06 JP   06A1
06C2 22 1A 01 LD   (011A),HL
06C5 CD 46 07 CALL 0746
06C8 79       LD   A,C
06C9 FE 0D    CP   0D
06CB FA D0 06 JP   N,06D0
06CE 0E 0C    LD   C,0C
06D0 2A 1A 01 LD   HL,(011A)
06D3 11 3F 02 LD   DE,023F
06D6 CD 54 0C CALL 0C54
06D9 EB       EX   DE,(HL)
06DA 2B       DEC  HL
06DB 3E 0D    LD   A,0D
06DD 77       LD   (HL),A
06DE 21 3F 02 LD   HL,023F
06E1 11 8B 02 LD   DE,028B
06E4 CD FB 06 CALL 06FB
06E7 B7       OR   A
06E8 C2 F4 06 JP   NZ,06F4
06EB 2A 1A 01 LD   HL,(011A)
06EE CD 46 07 CALL 0746
```

```
06F1 C3 A4 06 JP   06A4
06F4 2A 1A 01 LD   HL,(011A)
06F7 CD 46 07 CALL 0746
06FA C9       RET
06FB 7E       LD   A,(HL)
06FC FE 0D    CP   0D
06FE CA 13 07 JP   Z,0713
0701 4F       LD   C,A
0702 23       INC  HL
0703 EB       EX   DE,(HL)
0704 7E       LD   A,(HL)
0705 FE 0D    CP   0D
0707 CA 23 07 JP   Z,0723
070A B9       CP   C
070B C2 20 07 JP   NZ,0720
070E 23       INC  HL
070F EB       EX   DE,(HL)
0710 C3 FB 06 JP   06FB
0713 EB       EX   DE,(HL)
0714 7E       LD   A,(HL)
0715 FE 0D    CP   0D
0717 CA 29 07 JP   Z,0729
071A CD 46 07 CALL 0746
071D EB       EX   DE,(HL)
071E AF       XOR  A
071F C9       RET
0720 CD 46 07 CALL 0746
0723 EB       EX   DE,(HL)
0724 CD 46 07 CALL 0746
0727 AF       XOR  A
0728 C9       RET
0729 EB       EX   DE,(HL)
072A B4       OR   H
072B C9       RET
072C 7E       LD   A,(HL)
072D FE 20    CP   20
072F C0       RET  NZ
0730 23       INC  HL
0731 C3 2C 07 JP   072C
0734 7E       LD   A,(HL)
0735 FE 20    CP   20
0737 C0       RET  NZ
0738 2B       DEC  HL
0739 C3 34 07 JP   0734
073C CD 2C 07 CALL 072C
073F 11 8B 02 LD   DE,028B
0742 CD 54 07 CALL 0754
0745 C9       RET
0746 0E 01    LD   C,01
0748 CD B9 07 CALL 07B9
074B C8       RET  Z
074C 79       LD   A,C
074D C6 01    ADD  01
074F 4F       LD   C,A
0750 23       INC  HL
0751 C3 48 07 JP   0748
0754 0E 01    LD   C,01
0756 CD B9 07 CALL 07B9
```

```
0759 CA 68 07 JP   Z,0768
075C 23       INC  HL
075D EB       EX   DE,(HL)
075E 70       LD   (HL),B
075F 79       LD   A,C
0760 C6 01    ADD  01
0762 4F       LD   C,A
0763 23       INC  HL
0764 EB       EX   DE,(HL)
0765 C3 56 07 JP   0756
0768 23       INC  HL
0769 EB       EX   DE,(HL)
076A 3E 0D    LD   A,0D
076C 77       LD   (HL),A
076D 23       INC  HL
076E EB       EX   DE,(HL)
076F AF       XOR  A
0770 C9       RET
0771 11 8B 02 LD   DE,028B
0774 CD 54 07 CALL 0754
0777 79       LD   A,C
0778 FE 03    CP   03
077A F8       RET  M
077B 62       LD   H,D
077C 6B       LD   L,E
077D 2B       DEC  HL
077E 2B       DEC  HL
077F CD 34 07 CALL 0734
0782 FE 59    CP   59
0784 CA 8A 07 JP   Z,078A
0787 FE 4E    CP   4E
0789 C0       RET  NZ
078A 3E 0D    LD   A,0D
078C 77       LD   (HL),A
078D C9       RET
078E 06 3A    LD   B,3A
0790 CD 37 08 CALL 0837
0793 2A 22 01 LD   HL,(0122)
0796 11 4B 02 LD   DE,024B
0799 CD 54 0C CALL 0C54
079C C9       RET
079D 0E 01    LD   C,01
079F 7E       LD   A,(HL)
07A0 FE 0D    CP   0D
07A2 C8       RET  Z
07A3 FE 01    CP   01
07A5 C8       RET  Z
07A6 79       LD   A,C
07A7 C6 01    ADD  01
07A9 4F       LD   C,A
07AA 23       INC  HL
07AB C3 9F 07 JP   079F
07AE 79       LD   A,C
07AF FE 01    CP   01
07B1 C8       RET  Z
07B2 D6 01    SUB  01
07B4 4F       LD   C,A
07B5 2B       DEC  HL
```

```
07B6 C3 AE 07 JP   07AE
07B9 7E       LD   A,(HL)
07BA 47       LD   B,A
07BB FE 20    CP   20
07BD C8       RET  Z
07BE FE 0D    CP   0D
07C0 C8       RET  Z
07C1 FE 2C    CP   2C
07C3 C8       RET  Z
07C4 FE 3B    CP   3B
07C6 C8       RET  Z
07C7 FE 3A    CP   3A
07C9 C8       RET  Z
07CA FE 2E    CP   2E
07CC C8       RET  Z
07CD FE 3F    CP   3F
07CF C8       RET  Z
07D0 FE 21    CP   21
07D2 C8       RET  Z
07D3 FE 22    CP   22
07D5 C8       RET  Z
07D6 FE 28    CP   28
07D8 C8       RET  Z
07D9 FE 29    CP   29
07DB C8       RET  Z
07DC FE 27    CP   27
07DE C8       RET  Z
07DF FE 01    CP   01
07E1 C9       RET
07E2 AF       XOR  A
07E3 32 48 01 LD   (0148),A
07E6 2A 1A 01 LD   HL,(011A)
07E9 EB       EX   DE,(HL)
07EA 2A 1C 01 LD   HL,(011C)
07ED 46       LD   B,(HL)
07EE 23       INC  HL
07EF EB       EX   DE,(HL)
07F0 CD 37 08 CALL 0837
07F3 79       LD   A,C
07F4 B7       OR   A
07F5 C8       RET  Z
07F6 22 1A 01 LD   (011A),HL
07F9 3A 48 01 LD   A,(0148)
07FC 81       ADD  C
07FD 32 48 01 LD   (0148),A
0800 2A 1C 01 LD   HL,(011C)
0803 CD 9D 07 CALL 079D
0806 79       LD   A,C
0807 D6 01    SUB  01
0809 4F       LD   C,A
080A 11 8B 02 LD   DE,028B
080D 2A 1A 01 LD   HL,(011A)
0810 CD 54 0C CALL 0C54
0813 22 36 01 LD   (0136),HL
0816 EB       EX   DE,(HL)
0817 3E 0D    LD   A,0D
0819 77       LD   (HL),A
```

```
081A 2A 1C 01 LD   HL,(011C)
081D EB       EX   DE,(HL)
081E 21 8B 02 LD   HL,028B
0821 CD FB 06 CALL 06FB
0824 B7       OR   A
0825 C2 32 08 JP   NZ,0832
0828 2A 1A 01 LD   HL,(011A)
082B 23       INC  HL
082C 22 1A 01 LD   (011A),HL
082F C3 E6 07 JP   07E6
0832 21 48 01 LD   HL,0148
0835 4E       LD   C,(HL)
0836 C9       RET
0837 0E 01    LD   C,01
0839 7E       LD   A,(HL)
083A B8       CP   B
083B C8       RET  Z
083C FE 0D    CP   0D
083E CA 49 08 JP   Z,0849
0841 79       LD   A,C
0842 C6 01    ADD  01
0844 4F       LD   C,A
0845 23       INC  HL
0846 C3 39 08 JP   0839
0849 AF       XOR  A
084A 4F       LD   C,A
084B C9       RET
084C 0E 01    LD   C,01
084E 7E       LD   A,(HL)
084F FE 24    CP   24
0851 C8       RET  Z
0852 FE 23    CP   23
0854 C8       RET  Z
0855 FE 0D    CP   0D
0857 C8       RET  Z
0858 79       LD   A,C
0859 C6 01    ADD  01
085B 4F       LD   C,A
085C 23       INC  HL
085D C3 4E 08 JP   084E
0860 7E       LD   A,(HL)
0861 46       LD   B,(HL)
0862 FE 41    CP   41
0864 FA 6E 08 JP   M,086E
0867 FE 5A    CP   5A
0869 F2 6E 08 JP   P,086E
086C AF       XOR  A
086D C9       RET
086E B4       OR   H
086F C9       RET
0870 CD 9D 07 CALL 079D
0873 41       LD   B,C
0874 CD AE 07 CALL 07AE
0877 48       LD   C,B
0878 C9       RET
0879 2A 1A 01 LD   HL,(011A)
087C 22 38 01 LD   (0138),HL
```

```
087F 22 22 01 LD   (0122),HL
0882 21 4C 01 LD   HL,014C
0885 22 20 01 LD   (0120),HL
0888 EB       EX   DE,(HL)
0889 2A 38 01 LD   HL,(0138)
088C CD 4C 08 CALL 084C
088F FE 24    CP   24
0891 CA 9C 08 JP   Z,089C
0894 FE 23    CP   23
0896 CA AD 08 JP   Z,08AD
0899 C3 B3 08 JP   08B3
089C CD 11 09 CALL 0911
089F FE 01    CP   01
08A1 C2 89 08 JP   NZ,0889
08A4 11 4C 01 LD   DE,014C
08A7 2A 22 01 LD   HL,(0122)
08AA C3 BA 08 JP   08BA
08AD CD CC 08 CALL 08CC
08B0 C3 9F 08 JP   089F
08B3 2A 20 01 LD   HL,(0120)
08B6 EB       EX   DE,(HL)
08B7 2A 38 01 LD   HL,(0138)
08BA CD 70 08 CALL 0870
08BD CD 54 0C CALL 0C54
08C0 3E 0D    LD   A,0D
08C2 EB       EX   DE,(HL)
08C3 77       LD   (HL),A
08C4 21 4C 01 LD   HL,014C
08C7 CD 49 0D CALL 0D49
08CA AF       XOR  A
08CB C9       RET
08CC 79       LD   A,C
08CD D6 01    SUB  01
08CF CA DD 08 JP   Z,08DD
08D2 4F       LD   C,A
08D3 2A 20 01 LD   HL,(0120)
08D6 EB       EX   DE,(HL)
08D7 2A 38 01 LD   HL,(0138)
08DA CD 54 0C CALL 0C54
08DD 44       LD   B,H
08DE 4D       LD   C,L
08DF EB       EX   DE,(HL)
08E0 22 20 01 LD   (0120),HL
08E3 60       LD   H,B
08E4 69       LD   L,C
08E5 23       INC  HL
08E6 4E       LD   C,(HL)
08E7 23       INC  HL
08E8 22 38 01 LD   (0138),HL
08EB 41       LD   B,C
08EC CD D8 04 CALL 04D8
08EF FE 01    CP   01
08F1 C8       RET  Z
08F2 23       INC  HL
08F3 5E       LD   E,(HL)
08F4 21 8B 02 LD   HL,028B
08F7 CD 65 06 CALL 0665
```

```
08FA 2A 20 01 LD   HL,(0120)
08FD EB       EX   DE,(HL)
08FE 21 8B 02 LD   HL,028B
0901 CD 70 08 CALL 0870
0904 79       LD   A,C
0905 D6 01    SUB  01
0907 4F       LD   C,A
0908 CD 54 0C CALL 0C54
090B EB       EX   DE,(HL)
090C 22 20 01 LD   (0120),HL
090F AF       XOR  A
0910 C9       RET
0911 79       LD   A,C
0912 D6 01    SUB  01
0914 CA 22 09 JP   Z,0922
0917 4F       LD   C,A
0918 2A 20 01 LD   HL,(0120)
091B EB       EX   DE,(HL)
091C 2A 38 01 LD   HL,(0138)
091F CD 54 0C CALL 0C54
0922 44       LD   B,H
0923 4D       LD   C,L
0924 EB       EX   DE,(HL)
0925 22 20 01 LD   (0120),HL
0928 60       LD   H,B
0929 69       LD   L,C
092A CD 3C 07 CALL 073C
092D 2B       DEC  HL
092E 22 38 01 LD   (0138),HL
0931 2A 34 01 LD   HL,(0134)
0934 23       INC  HL
0935 CD 96 06 CALL 0696
0938 FE 01    CP   01
093A C8       RET  Z
093B CD 3F 09 CALL 093F
093E C9       RET
093F 23       INC  HL
0940 44       LD   B,H
0941 4D       LD   C,L
0942 2A 20 01 LD   HL,(0120)
0945 EB       EX   DE,(HL)
0946 60       LD   H,B
0947 69       LD   L,C
0948 CD 70 08 CALL 0870
094B 79       LD   A,C
094C D6 01    SUB  01
094E 4F       LD   C,A
094F CD 54 0C CALL 0C54
0952 EB       EX   DE,(HL)
0953 22 20 01 LD   (0120),HL
0956 AF       XOR  A
0957 C9       RET
0958 2A 1A 01 LD   HL,(011A)
095B CD 2C 07 CALL 072C
095E FE 24    CP   24
0960 C2 70 09 JP   NZ,0970
0963 11 3F 02 LD   DE,023F
0966 CD 54 07 CALL 0754
```

```
0969 CD 88 09 CALL 0988
096C CD B3 09 CALL 09B3
096F C9       RET
0970 FE 23    CP   23
0972 C2 84 09 JP   NZ,0984
0975 23       INC  HL
0976 7E       LD   A,(HL)
0977 32 4A 01 LD   (014A),A
097A CD 88 09 CALL 0988
097D CD 9F 09 CALL 099F
0980 CD 05 0A CALL 0A05
0983 C9       RET
0984 CD 88 09 CALL 0988
0987 C9       RET
0988 CD 65 0C CALL 0C65
098B 21 9D 01 LD   HL,019D
098E 7E       LD   A,(HL)
098F FE 1A    CP   1A
0991 C0       RET  NZ
0992 23       INC  HL
0993 7E       LD   A,(HL)
0994 FE 0D    CP   0D
0996 CA 0F 03 JP   Z,030F
0999 CD CE 03 CALL 03CE
099C C3 88 09 JP   0988
099F FE 30    CP   30
09A1 FA A7 09 JP   M,09A7
09A4 FE 3A    CP   3A
09A6 F8       RET  M
09A7 21 4A 0E LD   HL,0E4A
09AA CD DE 0D CALL 0DDE
09AD CD 88 09 CALL 0988
09B0 C3 9F 09 JP   099F
09B3 21 9D 01 LD   HL,019D
09B6 CD 9D 07 CALL 079D
09B9 41       LD   B,C
09BA 2A 34 01 LD   HL,(0134)
09BD CD E6 09 CALL 09E6
09C0 79       LD   A,C
09C1 B7       OR   A
09C2 C8       RET  Z
09C3 21 9D 01 LD   HL,019D
09C6 CD 54 0C CALL 0C54
09C9 21 3F 02 LD   HL,023F
09CC CD 9D 07 CALL 079D
09CF 41       LD   B,C
09D0 2A 34 01 LD   HL,(0134)
09D3 CD E6 09 CALL 09E6
09D6 79       LD   A,C
09D7 B7       OR   A
09D8 C8       RET  Z
09D9 21 3F 02 LD   HL,023F
09DC CD 54 0C CALL 0C54
09DF 6B       LD   L,E
09E0 62       LD   H,D
09E1 2B       DEC  HL
09E2 3E 20    LD   A,20
```

```
09E4 77          LD   (HL),A
09E5 C9          RET
09E6 5D          LD   E,L
09E7 54          LD   D,H
09E8 2B          DEC  HL
09E9 7E          LD   A,(HL)
09EA FE 01       CP   01
09EC CA FC 09 JP   Z,09FC
09EF 78          LD   A,B
09F0 D6 01       SUB  01
09F2 47          LD   B,A
09F3 C2 E6 09 JP   NZ,09E6
09F6 22 34 01 LD   (0134),HL
09F9 36 01       LD   (HL),01
09FB C9          RET
09FC 21 FF 0D LD   HL,0DFF
09FF CD DE 0D CALL 0DDE
0A02 0E 00       LD   C,00
0A04 C9          RET
0A05 21 9D 01 LD   HL,019D
0A08 CD 2C 07 CALL 072C
0A0B CD 31 06 CALL 0631
0A0E 21 4A 01 LD   HL,014A
0A11 46          LD   B,(HL)
0A12 CD D8 04 CALL 04D8
0A15 FE 01       CP   01
0A17 CC CB 04 CALL Z,04CB
0A1A C8          RET  Z
0A1B 23          INC  HL
0A1C 73          LD   (HL),E
0A1D C9          RET
0A1E 21 47 01 LD   HL,0147
0A21 36 5E       LD   (HL),5E
0A23 C3 2B 0A JP   0A2B
0A26 21 47 01 LD   HL,0147
0A29 36 2C       LD   (HL),2C
0A2B 2A 1A 01 LD   HL,(011A)
0A2E 22 3C 01 LD   (013C),HL
0A31 CD 80 0A CALL 0A80
0A34 2A 3C 01 LD   HL,(013C)
0A37 22 1A 01 LD   (011A),HL
0A3A CD D7 0A CALL 0AD7
0A3D 21 9D 01 LD   HL,019D
0A40 22 1A 01 LD   (011A),HL
0A43 CD A9 0A CALL 0AA9
0A46 CD D7 0A CALL 0AD7
0A49 21 9D 01 LD   HL,019D
0A4C 22 1A 01 LD   (011A),HL
0A4F 21 EE 01 LD   HL,01EE
0A52 22 1C 01 LD   (011C),HL
0A55 CD E2 07 CALL 07E2
0A58 79          LD   A,C
0A59 B7          OR   A
0A5A C4 76 0A CALL NZ,0A76
0A5D C0          RET  NZ
0A5E 2A 3C 01 LD   HL,(013C)
0A61 2B          DEC  HL
0A62 7E          LD   A,(HL)
```

```
0A63 FE 0D    CP   0D
0A65 CA 72 0A JP   Z,0A72
0A68 23       INC  HL
0A69 7E       LD   A,(HL)
0A6A FE 0D    CP   0D
0A6C CA 72 0A JP   Z,0A72
0A6F C3 31 0A JP   0A31
0A72 CD 7B 0A CALL 0A7B
0A75 C9       RET
0A76 B4       OR   H
0A77 32 4B 01 LD   (014B),A
0A7A C9       RET
0A7B AF       XOR  A
0A7C 32 4B 01 LD   (014B),A
0A7F C9       RET
0A80 2A 3C 01 LD   HL,(013C)
0A83 11 EE 01 LD   DE,01EE
0A86 CD 9F 0A CALL 0A9F
0A89 CA 94 0A JP   Z,0A94
0A8C 23       INC  HL
0A8D EB       EX   DE,(HL)
0A8E 70       LD   (HL),B
0A8F 23       INC  HL
0A90 EB       EX   DE,(HL)
0A91 C3 86 0A JP   0A86
0A94 23       INC  HL
0A95 EB       EX   DE,(HL)
0A96 3E 0D    LD   A,0D
0A98 77       LD   (HL),A
0A99 23       INC  HL
0A9A EB       EX   DE,(HL)
0A9B 22 3C 01 LD   (013C),HL
0A9E C9       RET
0A9F 7E       LD   A,(HL)
0AA0 47       LD   B,A
0AA1 FE 0D    CP   0D
0AA3 C8       RET  Z
0AA4 3A 47 01 LD   A,(0147)
0AA7 B8       CP   B
0AA8 C9       RET
0AA9 21 8B 02 LD   HL,028B
0AAC 3E 20    LD   A,20
0AAE 77       LD   (HL),A
0AAF 23       INC  HL
0AB0 EB       EX   DE,(HL)
0AB1 2A 1A 01 LD   HL,(011A)
0AB4 CD 9D 07 CALL 079D
0AB7 2A 1A 01 LD   HL,(011A)
0ABA CD 54 0C CALL 0C54
0ABD EB       EX   DE,(HL)
0ABE 2B       DEC  HL
0ABF 3E 20    LD   A,20
0AC1 77       LD   (HL),A
0AC2 23       INC  HL
0AC3 3E 0D    LD   A,0D
0AC5 77       LD   (HL),A
0AC6 2A 1A 01 LD   HL,(011A)
0AC9 EB       EX   DE,(HL)
```

```
0ACA 21 8B 02 LD   HL,028B
0ACD CD 9D 07 CALL 079D
0AD0 21 8B 02 LD   HL,028B
0AD3 CD 54 0C CALL 0C54
0AD6 C9       RET
0AD7 2A 1A 01 LD   HL,(011A)
0ADA 22 1C 01 LD   (011C),HL
0ADD 7E       LD   A,(HL)
0ADE FE 0D    CP   0D
0AE0 C8       RET  Z
0AE1 FE 20    CP   20
0AE3 CA EA 0A JP   Z,0AEA
0AE6 23       INC  HL
0AE7 C3 DA 0A JP   0ADA
0AEA 23       INC  HL
0AEB 7E       LD   A,(HL)
0AEC FE 20    CP   20
0AEE C2 DE 0A JP   NZ,0ADE
0AF1 22 1C 01 LD   (011C),HL
0AF4 CD FD 0A CALL 0AFD
0AF7 2A 1C 01 LD   HL,(011C)
0AFA C3 EB 0A JP   0AEB
0AFD 23       INC  HL
0AFE 46       LD   B,(HL)
0AFF 2B       DEC  HL
0B00 70       LD   (HL),B
0B01 7E       LD   A,(HL)
0B02 FE 0D    CP   0D
0B04 C8       RET  Z
0B05 23       INC  HL
0B06 C3 FD 0A JP   0AFD
0B09 2A 1A 01 LD   HL,(011A)
0B0C 06 3D    LD   B,3D
0B0E CD 37 0B CALL 0B37
0B11 79       LD   A,C
0B12 B7       OR   A
0B13 CA 77 0B JP   Z,0B77
0B16 23       INC  HL
0B17 CD 2C 07 CALL 072C
0B1A FE 2D    CP   2D
0B1C C2 2C 0B JP   NZ,0B2C
0B1F 22 1C 01 LD   (011C),HL
0B22 AF       XOR  A
0B23 32 49 01 LD   (0149),A
0B26 2A 1C 01 LD   HL,(011C)
0B29 C3 60 0B JP   0B60
0B2C CD 9F 0B CALL 0B9F
0B2F 21 49 01 LD   HL,0149
0B32 73       LD   (HL),E
0B33 2A 38 01 LD   HL,(0138)
0B36 23       INC  HL
0B37 CD 2C 07 CALL 072C
0B3A FE 0D    CP   0D
0B3C CA 73 0B JP   Z,0B73
0B3F FE 2D    CP   2D
0B41 CA 60 0B JP   Z,0B60
0B44 FE 2B    CP   2B
0B46 CA 4F 0B JP   Z,0B4F
```

```
0B49 CD 9F 0B CALL 0B9F
0B4C C3 73 0B JP   0B73
0B4F 23       INC  HL
0B50 CD 9F 0B CALL 0B9F
0B53 3A 49 01 LD   A,(0149)
0B56 83       ADD  E
0B57 FE 64    CP   64
0B59 F2 81 0B JP   P,0B81
0B5C 5F       LD   E,A
0B5D C3 73 0B JP   0B73
0B60 23       INC  HL
0B61 CD 9F 0B CALL 0B9F
0B64 3A 49 01 LD   A,(0149)
0B67 93       SUB  E
0B68 FE 9D    CP   9D
0B6A F2 72 0B JP   P,0B72
0B6D FE 64    CP   64
0B6F F2 90 0B JP   P,0B90
0B72 5F       LD   E,A
0B73 CD C0 0B CALL 0BC0
0B76 C9       RET
0B77 CD 79 08 CALL 0879
0B7A 21 08 0E LD   HL,0E08
0B7D CD DE 0D CALL 0DDE
0B80 C9       RET
0B81 1E 63    LD   E,63
0B83 CD C0 0B CALL 0BC0
0B86 CD 79 08 CALL 0879
0B89 21 1C 0E LD   HL,0E1C
0B8C CD DE 0D CALL 0DDE
0B8F C9       RET
0B90 1E 9D    LD   E,9D
0B92 CD C0 0B CALL 0BC0
0B95 CD 79 08 CALL 0879
0B98 21 29 0E LD   HL,0E29
0B9B CD DE 0D CALL 0DDE
0B9E C9       RET
0B9F CD 2C 07 CALL 072C
0BA2 5F       LD   E,A
0BA3 CD 60 08 CALL 0860
0BA6 CA B0 0B JP   Z,0BB0
0BA9 CD 31 06 CALL 0631
0BAC 22 38 01 LD   (0138),HL
0BAF C9       RET
0BB0 22 38 01 LD   (0138),HL
0BB3 46       LD   B,(HL)
0BB4 CD D8 04 CALL 04D8
0BB7 FE 01    CP   01
0BB9 CC CB 04 CALL Z,04CB
0BBC C8       RET  Z
0BBD 23       INC  HL
0BBE 5E       LD   E,(HL)
0BBF C9       RET
0BC0 2A 1A 01 LD   HL,(011A)
0BC3 CD 2C 07 CALL 072C
0BC6 CD 60 08 CALL 0860
0BC9 C2 77 0B JP   NZ,0B77
0BCC 46       LD   B,(HL)
```

```
0BCD CD D8 04 CALL 04D8
0BD0 FE 01    CP   01
0BD2 CC CB 04 CALL Z,04CB
0BD5 C8       RET  Z
0BD6 23       INC  HL
0BD7 73       LD   (HL),E
0BD8 C9       RET
0BD9 B4       OR   H
0BDA C9       RET
0BDB C9       RET
0BDC 3A 45 01 LD   A,(0145)
0BDF B7       OR   A
0BE0 CA 0F 03 JP   Z,030F
0BE3 CD 38 0C CALL 0C38
0BE6 C9       RET
0BE7 CD 00 14 CALL 1400
0BEA CD 56 03 CALL 0356
0BED C9       RET
0BEE 2A 34 01 LD   HL,(0134)
0BF1 36 20    LD   (HL),20
0BF3 2A 3E 01 LD   HL,(013E)
0BF6 36 01    LD   (HL),01
0BF8 2B       DEC  HL
0BF9 22 34 01 LD   (0134),HL
0BFC C9       RET
0BFD 21 56 02 LD   HL,0256
0C00 06 41    LD   B,41
0C02 78       LD   A,B
0C03 FE 5B    CP   5B
0C05 C8       RET  Z
0C06 77       LD   (HL),A
0C07 23       INC  HL
0C08 3E 00    LD   A,00
0C0A 77       LD   (HL),A
0C0B 23       INC  HL
0C0C 78       LD   A,B
0C0D C6 01    ADD  01
0C0F 47       LD   B,A
0C10 C3 03 0C JP   0C03
0C13 21 45 01 LD   HL,0145
0C16 7E       LD   A,(HL)
0C17 C6 01    ADD  01
0C19 FE 08    CP   08
0C1B FA 25 0C JP   N,0C25
0C1E 21 36 0E LD   HL,0E36
0C21 CD DE 0D CALL 0DDE
0C24 C9       RET
0C25 77       LD   (HL),A
0C26 87       ADD  A
0C27 4F       LD   C,A
0C28 2A 3A 01 LD   HL,(013A)
0C2B EB       EX   DE,(HL)
0C2C 21 24 01 LD   HL,0124
0C2F 7D       LD   A,L
0C30 81       ADD  C
0C31 6F       LD   L,A
0C32 72       LD   (HL),D
0C33 C6 01    ADD  01
```

```
0C35 6F          LD   L,A
0C36 73          LD   (HL),E
0C37 C9          RET
0C38 3A 45 01 LD   A,(0145)
0C3B 87          ADD  A
0C3C 4F          LD   C,A
0C3D 21 24 01 LD   HL,0124
0C40 7D          LD   A,L
0C41 81          ADD  C
0C42 6F          LD   L,A
0C43 56          LD   D,(HL)
0C44 C6 01       ADD  01
0C46 6F          LD   L,A
0C47 5E          LD   E,(HL)
0C48 EB          EX   DE,(HL)
0C49 22 3A 01 LD   (013A),HL
0C4C 21 45 01 LD   HL,0145
0C4F 7E          LD   A,(HL)
0C50 D6 01       SUB  01
0C52 77          LD   (HL),A
0C53 C9          RET
0C54 79          LD   A,C
0C55 B7          OR   A
0C56 C8          RET  Z
0C57 46          LD   B,(HL)
0C58 23          INC  HL
0C59 EB          EX   DE,(HL)
0C5A 70          LD   (HL),B
0C5B 23          INC  HL
0C5C EB          EX   DE,(HL)
0C5D 79          LD   A,C
0C5E D6 01       SUB  01
0C60 4F          LD   C,A
0C61 C2 54 0C JP   NZ,0C54
0C64 C9          RET
0C65 21 9D 01 LD   HL,019D
0C68 22 36 01 LD   (0136),HL
0C6B 21 44 01 LD   HL,0144
0C6E 46          LD   B,(HL)
0C6F 2A 36 01 LD   HL,(0136)
0C72 CD 00 01 CALL 0100
0C75 FE 5F       CP   5F
0C77 CA B0 0C JP   Z,0CB0
0C7A FE 7F       CP   7F
0C7C CA B0 0C JP   Z,0CB0
0C7F FE 15       CP   15
0C81 CA A5 0C JP   Z,0CA5
0C84 FE 61       CP   61
0C86 FA 8B 0C JP   N,0C8B
0C89 EE 20       XOR  20
0C8B 77          LD   (HL),A
0C8C 23          INC  HL
0C8D 22 36 01 LD   (0136),HL
0C90 FE 0D       CP   0D
0C92 CA A1 0C JP   Z,0CA1
0C95 05          DEC  B
0C96 78          LD   A,B
```

```
0C97 B7          OR   A
0C98 C2 6F 0C JP   NZ,0C6F
0C9B 0E 0D       LD   C,0D
0C9D 71          LD   (HL),C
0C9E CD 03 01 CALL 0103
0CA1 CD C9 0C CALL 0CC9
0CA4 C9          RET
0CA5 0E 3C       LD   C,3C
0CA7 CD 03 01 CALL 0103
0CAA CD C4 0C CALL 0CC4
0CAD C3 65 0C JP   0C65
0CB0 78          LD   A,B
0CB1 21 44 01 LD   HL,0144
0CB4 4E          LD   C,(HL)
0CB5 B9          CP   C
0CB6 CA 6F 0C JP   Z,0C6F
0CB9 04          INC  B
0CBA 2A 36 01 LD   HL,(0136)
0CBD 2B          DEC  HL
0CBE 22 36 01 LD   (0136),HL
0CC1 C3 72 0C JP   0C72
0CC4 00          NOP
0CC5 00          NOP
0CC6 00          NOP
0CC7 00          NOP
0CC8 00          NOP
0CC9 0E 0D       LD   C,0D
0CCB CD 03 01 CALL 0103
0CCE C9          RET
0CCF CD 18 06 CALL 0618
0CD2 21 00 15 LD   HL,1500
0CD5 7E          LD   A,(HL)
0CD6 22 22 01 LD   (0122),HL
0CD9 FE 01       CP   01
0CDB CA 2F 0D JP   Z,0D2F
0CDE CD 06 01 CALL 0106
0CE1 FE 20       CP   20
0CE3 C2 F2 0C JP   NZ,0CF2
0CE6 C3 D5 0C JP   0CD5
0CE9 7E          LD   A,(HL)
0CEA FE 01       CP   01
0CEC CA 2F 0D JP   Z,0D2F
0CEF CD 06 01 CALL 0106
0CF2 FE 00       CP   00
0CF4 CA EF 0C JP   Z,0CEF
0CF7 FE 7F       CP   7F
0CF9 CA EF 0C JP   Z,0CEF
0CFC FE 1A       CP   1A
0CFE CA 3B 0D JP   Z,0D3B
0D01 FE 15       CP   15
0D03 CA 21 0D JP   Z,0D21
0D06 FE 5F       CP   5F
0D08 CA 1D 0D JP   Z,0D1D
0D0B 77          LD   (HL),A
0D0C 23          INC  HL
0D0D FE 0D       CP   0D
0D0F C2 E9 0C JP   NZ,0CE9
0D12 CD C9 0C CALL 0CC9
```

```
0D15 C3 D5 0C JP   0CD5
0D18 77       LD   (HL),A
0D19 23       INC  HL
0D1A C3 E9 0C JP   0CE9
0D1D 2B       DEC  HL
0D1E C3 EF 0C JP   0CEF
0D21 0E 3C    LD   C,3C
0D23 CD 03 01 CALL 0103
0D26 CD C4 0C CALL 0CC4
0D29 2A 22 01 LD   HL,(0122)
0D2C C3 EF 0C JP   0CEF
0D2F 2A 22 01 LD   HL,(0122)
0D32 CD 49 0D CALL 0D49
0D35 21 F5 0D LD   HL,0DF5
0D38 CD DE 0D CALL 0DDE
0D3B 36 01    LD   (HL),01
0D3D 22 18 01 LD   (0118),HL
0D40 CD C4 0C CALL 0CC4
0D43 21 45 01 LD   HL,0145
0D46 36 00    LD   (HL),00
0D48 C9       RET
0D49 7E       LD   A,(HL)
0D4A 23       INC  HL
0D4B 54       LD   D,H
0D4C 5D       LD   E,L
0D4D FE 01    CP   01
0D4F C8       RET  Z
0D50 4F       LD   C,A
0D51 2A 40 01 LD   HL,(0140)
0D54 CD 6D 0D CALL 0D6D
0D57 FE 0D    CP   0D
0D59 CA 60 0D JP   Z,0D60
0D5C EB       EX   DE,(HL)
0D5D C3 49 0D JP   0D49
0D60 0E 0A    LD   C,0A
0D62 2A 40 01 LD   HL,(0140)
0D65 CD 6D 0D CALL 0D6D
0D68 EB       EX   DE,(HL)
0D69 CD 7A 04 CALL 047A
0D6C C9       RET
0D6D E9       JP   (HL)
0D6E 21 00 15 LD   HL,1500
0D71 CD 49 0D CALL 0D49
0D74 FE 01    CP   01
0D76 C2 71 0D JP   NZ,0D71
0D79 C9       RET
0D7A 21 03 01 LD   HL,0103
0D7D 22 40 01 LD   (0140),HL
0D80 CD 6E 0D CALL 0D6E
0D83 C9       RET
0D84 21 09 01 LD   HL,0109
0D87 22 40 01 LD   (0140),HL
0D8A CD 6E 0D CALL 0D6E
0D8D 21 03 01 LD   HL,0103
0D90 22 40 01 LD   (0140),HL
0D93 C9       RET
0D94 21 0C 01 LD   HL,010C
```

```
0D97 22 40 01 LD   (0140),HL
0D9A CD 6E 0D CALL 0D6E
0D9D 21 03 01 LD   HL,0103
0DA0 22 40 01 LD   (0140),HL
0DA3 C9       RET
0DA4 79       LD   A,C
0DA5 C3 00 0F JP   0F00
0DA8 00       NOP
0DA9 00       NOP
0DAA 00       NOP
0DAB 00       NOP
0DAC 00       NOP
0DAD 00       NOP
0DAE 00       NOP
0DAF 00       NOP
0DB0 00       NOP
0DB1 00       NOP
0DB2 00       NOP
0DB3 00       NOP
0DB4 00       NOP
0DB5 00       NOP
0DB6 00       NOP
0DB7 00       NOP
0DB8 00       NOP
0DB9 00       NOP
0DBA 00       NOP
0DBB 00       NOP
0DBC 00       NOP
0DBD 00       NOP
0DBE 00       NOP
0DBF 00       NOP
0DC0 00       NOP
0DC1 00       NOP
0DC2 00       NOP
0DC3 00       NOP
0DC4 00       NOP
0DC5 00       NOP
0DC6 00       NOP
0DC7 00       NOP
0DC8 00       NOP
0DC9 00       NOP
0DCA 00       NOP
0DCB 00       NOP
0DCC 00       NOP
0DCD C9       RET
0DCE 00       NOP
0DCF 00       NOP
0DD0 00       NOP
0DD1 00       NOP
0DD2 00       NOP
0DD3 00       NOP
0DD4 00       NOP
0DD5 00       NOP
0DD6 00       NOP
0DD7 00       NOP
0DD8 00       NOP
0DD9 00       NOP
0DDA 00       NOP
```

```
0DDB 00       NOP
0DDC 00       NOP
0DDD C9       RET
0DDE CD 49 0D CALL 0D49
0DE1 AF       XOR  A
0DE2 C9       RET
0DE3 2D       DEC  L
0DE4 20 4C    JR   NZ,4C        *0E32*
0DE6 41       LD   B,C
0DE7 42       LD   B,D
0DE8 45       LD   B,L
0DE9 4C       LD   C,H
0DEA 20 4E    JR   NZ,4E        *0E3A*
0DEC 4F       LD   C,A
0DED 54       LD   D,H
0DEE 20 46    JR   NZ,46        *0E36*
0DF0 4F       LD   C,A
0DF1 55       LD   D,L
0DF2 4E       LD   C,(HL)
0DF3 44       LD   B,H
0DF4 0D       DEC  C
0DF5 2F       CPL
0DF6 4F       LD   C,A
0DF7 56       LD   D,(HL)
0DF8 45       LD   B,L
0DF9 52       LD   D,D
0DFA 46       LD   B,(HL)
0DFB 4C       LD   C,H
0DFC 4F       LD   C,A
0DFD 57       LD   D,A
0DFE 0D       DEC  C
0DFF 2A 4E 4F LD   HL,(4F4E)
0E02 20 52    JR   NZ,52        *0E56*
0E04 4F       LD   C,A
0E05 4F       LD   C,A
0E06 4D       LD   C,L
0E07 0D       DEC  C
0E08 2A 49 4C LD   HL,(4C49)
0E0B 4C       LD   C,H
0E0C 45       LD   B,L
0E0D 47       LD   B,A
0E0E 41       LD   B,C
0E0F 4C       LD   C,H
0E10 20 45    JR   NZ,45        *0E57*
0E12 58       LD   E,B
0E13 50       LD   D,B
0E14 52       LD   D,D
0E15 45       LD   B,L
0E16 53       LD   D,E
0E17 53       LD   D,E
0E18 49       LD   C,C
0E19 4F       LD   C,A
0E1A 4E       LD   C,(HL)
0E1B 0D       DEC  C
0E1C 2A 56 41 LD   HL,(4156)
0E1F 4C       LD   C,H
0E20 55       LD   D,L
0E21 45       LD   B,L
0E22 20 3E    JR   NZ,3E        *0E62*
```

```
0E24 20 20     JR   NZ,20       *0E46*
0E26 39        ADD  HL,SP
0E27 39        ADD  HL,SP
0E28 0D        DEC  C
0E29 2A 56 41  LD   HL,(4156)
0E2C 4C        LD   C,H
0E2D 55        LD   D,L
0E2E 45        LD   B,L
0E2F 20 3C     JR   NZ,3C       *0E6D*
0E31 20 2D     JR   NZ,2D       *0E60*
0E33 39        ADD  HL,SP
0E34 39        ADD  HL,SP
0E35 0D        DEC  C
0E36 2A 55 53  LD   HL,(5355)
0E39 45        LD   B,L
0E3A 20 44     JR   NZ,44       *0E80*
0E3C 45        LD   B,L
0E3D 50        LD   D,B
0E3E 54        LD   D,H
0E3F 48        LD   C,B
0E40 20 45     JR   NZ,45       *0E87*
0E42 58        LD   E,B
0E43 43        LD   B,E
0E44 45        LD   B,L
0E45 45        LD   B,L
0E46 44        LD   B,H
0E47 45        LD   B,L
0E48 44        LD   B,H
0E49 0D        DEC  C
0E4A 2A 4E 55  LD   HL,(554E)
0E4D 4D        LD   C,L
0E4E 45        LD   B,L
0E4F 52        LD   D,D
0E50 49        LD   C,C
0E51 43        LD   B,E
0E52 20 52     JR   NZ,52       *0EA6*
0E54 45        LD   B,L
0E55 53        LD   D,E
0E56 50        LD   D,B
0E57 4F        LD   C,A
0E58 4E        LD   C,(HL)
0E59 53        LD   D,E
0E5A 45        LD   B,L
0E5B 20 52     JR   NZ,52       *0EAF*
0E5D 45        LD   B,L
0E5E 51        LD   D,C
0E5F 55        LD   D,L
0E60 49        LD   C,C
0E61 52        LD   D,D
0E62 45        LD   B,L
0E63 44        LD   B,H
0E64 0D        DEC  C
0E65 2A 49 4E  LD   HL,(4E49)
0E68 54        LD   D,H
0E69 45        LD   B,L
0E6A 52        LD   D,D
0E6B 52        LD   D,D
0E6C 55        LD   D,L
```

```
0E6D 50          LD   D,B
0E6E 54          LD   D,H
0E6F 45          LD   B,L
0E70 44          LD   B,H
0E71 0D          DEC  C
0E72 54          LD   D,H
0E73 3A 0D 54 LD   A,(540D)
0E76 3A 50 49 LD   A,(4950)
0E79 4C          LD   C,H
0E7A 4F          LD   C,A
0E7B 54          LD   D,H
0E7C 20 5A       JR   NZ,5A        *0ED8*
0E7E 20 38       JR   NZ,38        *0EB8*
0E80 30 20       JR   NC,20        *0EA2*
0E82 20 31       JR   NZ,31        *0EB5*
0E84 2E 31       LD   L,31
0E86 8D          ADC  L
0E87 CC CF C1 CALL Z,C1CF
0E8A C4 A0 BA CALL NZ,BAA0
0E8D CC D3 CE CALL Z,CED3
0E90 B1          OR   C
0E91 00          NOP
0E92 49          LD   C,C
0E93 45          LD   B,L
0E94 50          LD   D,B
0E95 3A 0D 01 LD   A,(010D)
0E98 20 20       JR   NZ,20        *0EBA*
0E9A 20 20       JR   NZ,20        *0EBC*
0E9C 20 20       JR   NZ,20        *0EBE*
0E9E 20 20       JR   NZ,20        *0EC0*
0EA0 20 20       JR   NZ,20        *0EC2*
0EA2 20 20       JR   NZ,20        *0EC4*
0EA4 20 20       JR   NZ,20        *0EC6*
0EA6 20 20       JR   NZ,20        *0EC8*
0EA8 20 01       JR   NZ,01        *0EAB*
0EAA 54          LD   D,H
0EAB 3A 45 4E LD   A,(4E45)
0EAE 54          LD   D,H
0EAF 45          LD   B,L
0EB0 52          LD   D,D
0EB1 20 50       JR   NZ,50        *0F03*
0EB3 49          LD   C,C
0EB4 4C          LD   C,H
0EB5 4F          LD   C,A
0EB6 54          LD   D,H
0EB7 20 50       JR   NZ,50        *0F09*
0EB9 52          LD   D,D
0EBA 4F          LD   C,A
0EBB 47          LD   B,A
0EBC 52          LD   D,D
0EBD 41          LD   B,C
0EBE 4D          LD   C,L
0EBF 0D          DEC  C
0EC0 3A 54 45 LD   A,(4554)
0EC3 52          LD   D,D
0EC4 4D          LD   C,L
0EC5 49          LD   C,C
```

```
0EC6 4E        LD   C,(HL)
0EC7 41        LD   B,C
0EC8 54        LD   D,H
0EC9 45        LD   B,L
0ECA 20 49     JR   NZ,49       *0F15*
0ECC 4E        LD   C,(HL)
0ECD 50        LD   D,B
0ECE 55        LD   D,L
0ECF 54        LD   D,H
0ED0 20 57     JR   NZ,57       *0F29*
0ED2 49        LD   C,C
0ED3 54        LD   D,H
0ED4 48        LD   C,B
0ED5 20 43     JR   NZ,43       *0F1A*
0ED7 54        LD   D,H
0ED8 4C        LD   C,H
0ED9 2F        CPL
0EDA 5A        LD   E,D
0EDB 0D        DEC  C
0EDC 4C        LD   C,H
0EDD 4F        LD   C,A
0EDE 41        LD   B,C
0EDF 44        LD   B,H
0EE0 3A 0D 2A  LD   A,(2A0D)
0EE3 25        DEC  H
0EE4 20 49     JR   NZ,49       *0F2F*
0EE6 45        LD   B,L
0EE7 50        LD   D,B
0EE8 3A 0D 00  LD   A,(000D)
0EEB 00        NOP
0EEC 00        NOP
0EED 00        NOP
0EEE 00        NOP
0EEF 00        NOP
0EF0 00        NOP
0EF1 00        NOP
0EF2 00        NOP
0EF3 00        NOP
0EF4 00        NOP
0EF5 00        NOP
0EF6 00        NOP
0EF7 00        NOP
0EF8 00        NOP
0EF9 00        NOP
0EFA 00        NOP
0EFB 00        NOP
0EFC 00        NOP
0EFD 00        NOP
0EFE 00        NOP
0EFF 00        NOP
0F00 F5        PUSH AF
0F01 C5        PUSH BC
0F02 D5        PUSH DE
0F03 E5        PUSH HL
0F04 F6 80     OR   80
0F06 FE DE     CP   DE
0F08 20 02     JR   NZ,02       *0F0C*
0F0A 3E 9C     LD   A,9C
```

```
0F0C FE 8D    CP   8D
0F0E 28 3A    JR   Z,3A       *0F4A*
0F10 FE 8A    CP   8A
0F12 28 2C    JR   Z,2C       *0F40*
0F14 FE 00    CP   00
0F16 28 28    JR   Z,28       *0F40*
0F18 FE 9B    CP   9B
0F1A 20 10    JR   NZ,10       *0F2C*
0F1C CD 91 0F CALL 0F91
0F1F 2A D0 0F LD   HL,(0FD0)
0F22 2B       DEC  HL
0F23 2B       DEC  HL
0F24 22 D8 0F LD   (0FD8),HL
0F27 CD 77 0F CALL 0F77
0F2A 18 14    JR   14       *0F40*
0F2C 2A D0 0F LD   HL,(0FD0)
0F2F F5       PUSH AF
0F30 3A D5 0F LD   A,(0FD5)
0F33 BC       CP   H
0F34 CC 5B 0F CALL Z,0F5B
0F37 F1       POP  AF
0F38 77       LD   (HL),A
0F39 23       INC  HL
0F3A 22 D0 0F LD   (0FD0),HL
0F3D CD B2 E6 CALL E6B2
0F40 E1       POP  HL
0F41 D1       POP  DE
0F42 C1       POP  BC
0F43 F1       POP  AF
0F44 00       NOP
0F45 C9       RET
0F46 3E 0D    LD   A,0D
0F48 18 B6    JR   B6       *0F00*
0F4A CD 5B 0F CALL 0F5B
0F4D 3A 44 0F LD   A,(0F44)
0F50 B7       OR   A
0F51 28 ED    JR   Z,ED       *0F40*
0F53 CD 91 0F CALL 0F91
0F56 CD 77 0F CALL 0F77
0F59 18 E5    JR   E5       *0F40*
0F5B CD 91 0F CALL 0F91
0F5E 2A D2 0F LD   HL,(0FD2)
0F61 EB       EX   DE,(HL)
0F62 2A D4 0F LD   HL,(0FD4)
0F65 4C       LD   C,H
0F66 21 40 10 LD   HL,1040
0F69 7E       LD   A,(HL)
0F6A 12       LD   (DE),A
0F6B 13       INC  DE
0F6C 23       INC  HL
0F6D 7C       LD   A,H
0F6E B9       CP   C
0F6F 20 F8    JR   NZ,F8       *0F69*
0F71 2A D6 0F LD   HL,(0FD6)
0F74 22 D0 0F LD   (0FD0),HL
0F77 3E 7F    LD   A,7F
0F79 CD B2 E6 CALL E6B2
0F7C 2A D0 0F LD   HL,(0FD0)
0F7F EB       EX   DE,(HL)
```

```
0F80 2A D2 0F LD   HL,(0FD2)
0F83 7E       LD   A,(HL)
0F84 CD B2 E6 CALL E6B2
0F87 23       INC  HL
0F88 7D       LD   A,L
0F89 BB       CP   E
0F8A 20 F7    JR   NZ,F7       *0F83*
0F8C 7C       LD   A,H
0F8D BA       CP   D
0F8E 20 F3    JR   NZ,F3       *0F83*
0F90 C9       RET
0F91 2A D4 0F LD   HL,(0FD4)
0F94 4C       LD   C,H
0F95 2A D0 0F LD   HL,(0FD0)
0F98 7C       LD   A,H
0F99 B9       CP   C
0F9A C8       RET  Z
0F9B 36 A0    LD   (HL),A0
0F9D CD B0 E6 CALL E6B0
0FA0 23       INC  HL
0FA1 18 F5    JR   F5       *0F98*
0FA3 3E DF    LD   A,DF
0FA5 CD B2 E6 CALL E6B2
0FA8 E5       PUSH HL
0FA9 D5       PUSH DE
0FAA CD 91 0F CALL 0F91
0FAD CD 77 0F CALL 0F77
0FB0 D1       POP  DE
0FB1 E1       POP  HL
0FB2 CD 6A E3 CALL E36A
0FB5 FE C0    CP   C0
0FB7 38 07    JR   C,07       *0FC0*
0FB9 00       NOP
0FBA 00       NOP
0FBB 00       NOP
0FBC 00       NOP
0FBD 00       NOP
0FBE E6 DF    AND  DF
0FC0 E6 7F    AND  7F
0FC2 FE FF    CP   FF
0FC4 C8       RET  Z
0FC5 FE 5D    CP   5D
0FC7 CA 00 00 JP   Z,0000
0FCA FE 0D    CP   0D
0FCC C8       RET  Z
0FCD C3 00 0F JP   0F00
0FD0 C0       RET  NZ
0FD1 13       INC  DE
0FD2 00       NOP
0FD3 10 00    DJNZ 00       *0FD5*
0FD5 14       INC  D
0FD6 C0       RET  NZ
0FD7 13       INC  DE
0FD8 FE 74    CP   74
0FDA 76       HALT
0FDB 75       LD   (HL),L
0FDC F6 76    OR   76
0FDE 7D       LD   A,L
0FDF 76       HALT
0FE0 00       NOP
0FE1 45       LD   B,L
0FE2 40       LD   B,B
```

```
0FE3 02          LD   (BC),A
0FE4 26 46       LD   H,46
0FE6 24          INC  H
0FE7 40          LD   B,B
0FE8 03          INC  BC
0FE9 21 03 02 LD   HL,0203
0FEC 27          DAA
0FED 23          INC  HL
0FEE 06 05       LD   B,05
0FF0 76          HALT
0FF1 F4 36 74 CALL P,7436
0FF4 76          HALT
0FF5 74          LD   (HL),H
0FF6 3C          INC  A
0FF7 76          HALT
0FF8 76          HALT
0FF9 34          INC  (HL)
0FFA B6          OR   (HL)
0FFB 7E          LD   A,(HL)
0FFC 74          LD   (HL),H
0FFD F6 7C       OR   7C
0FFF 76          HALT
```

APPENDIX D
ASSEMBLY LANGUAGE LISTING
OF
FILES INTERFACE FOR Z-80 PILOT

```
FUNCTION:
TO LOAD LESSON FILES FOR EXECUTION BY
THE PILOT INTERPRETER .


CALLING ROUTINE:
NAME : LOAD OF PILOT INTERPRETER
LOCATION : 0BE7H

PARAMETERS, INPUT: NAME OF FILE TO BE LOADED
            (POINTED TO BY PILOT INPUT BUFFER
             POINTER ,IPTR)
PARAMETERS, OUTPUT: PHIMON LOAD COMMAND AND
            FILE NAME IN FORMAT:
             "LO FILE-NAME" OR "LO#X FILE-NAME"

CALLED ROUTINES :
PHIMON LOAD OVERLAY
PHIMON TV SUBROUTINE
PHIMON STOP SUBROUTINE
PHIMON OVLAY SUBROUTINE

EXECUTION STEPS :

1.0 IF (FIRST TIME IN ROUTINE ) THEN STEP 2.0
          ELSE STEP 3.0

2.0    FIRST = FALSE
2.1 LOAD PHIMON "LOAD" OVERLAY BY CALLING "OVRLAY"
2.3 MODIFY "LOAD" OVERLAY INTO SUBROUTINE
2.4 MODIFY ERROR POINTERS OF LOAD OVERLAY TO
    POINT TO THIS CODE VICE PHIMON ERROR CODE

3.0 FORMAT "LOAD" COMMAND FOR PHIMON

4.0 MOVE "LOAD" COMMAND TO PHIMON COMMAND BUFFER

5.0 CALL "LOAD" SUBROUTINE (CREATED IN STEP 2 )

5.1 IF ERROR THEN STEP 8

6.0 CALL PHIMON "STOP" SUBROUTINE TO STOP
    THE CASSETTE DRIVE

7.0 RETURN  TO CALLING ROUTINE IN PILOT INTERPRETER

8.0 CALL PHIMON TV ROUTINE TO DISPLAY ERROR MESSAGE
8.1 JUMP TO PHIMON
```

```
       ST   024000
********** STEP 1.0 ***************************************
ENTRY  LD   HL,FIRST   SET HL TO LOCATION OF FLAG
       CP   (HL) IS IT SET TO 0 INDICATING WE HAVE
       JR   Z,GO BEEN HERE BEFORE , IF SO JUMP
********** STEP 2.0 ***************************************
       LD   A,0   SET FLAG (FIRST) TO ZERO TO
       LD   (FIRST),A  FLAG WE HAVE LOADED "LOAD" BEFORE
       LD   A,4   LOAD PHIMON "LOAD" OVERLAY
       CALL OVRLAY  INTO MEMORY .
       LD   A,0C8H   DESTROY 'GOOD' FLAG OF OVERLAY
       LD   (LOAD),A   TO PREVENT PHIMON FROM USING IT.
       LD   (347172),A CHANGE TO A SUBROUTINE BY ADDING
*                       'RETURN' AS LAST STATEMENT.
       LD   HL,ERRMSG  LOCATION OF LOAD ERROR ROUTINE
       LD   (347012),HL  SET ERROR POINTER TO ERRMSG
       LD   (347071),HL        "
       LD   (347115),HL        "
       LD   (347151),HL        "
       LD   (347042),HL        "
********** STEP 3.0 ***************************************
GO     LD   HL,(IPTR)     WHEN THIS ROUTINE IS CALLED
*                          IS CALLED , THE PILOT INPUT
*                           BUFFER POINTER IS AT THE 1ST
*                            CHAR PAST THE LOAD COMMAND
*                             IPTR IS THE INPUT BUF PT.
       DEC  HL          MOVE POINTER OVER C/R
       XOR  A SET TO ZERO TO MARK COMMAND END FOR PHIMON.
       LD   M,A  THIS IS REQUIRED FOR ALL PHIMON COMMANDS.
       LD   A,':'  PLACE MASK IN ACC A FOR LOCATING ":"
SCAN   DEC  HL
       SET  7,(HL) ALL BIT 7'S MUST BE HIGH FOR PHIMON CMD
       CP   (HL)   IS IT A COLON ?
       JR   NZ,SCAN  NO , CONTINUE PROCESSING.
       LD   A,'O'    FORM PHIMON
       LD   M,A        COMMAND BY
       DEC  HL           ADDING "LO" TO
       LD   A,'L'
       LD   M,A                "LOAD:". HL POINTS TO
*                                 START OF COMMAND STRING.
********** STEP 4.0 ***************************************
       LD   DE,LINBUF DESTINATION
       LD   BC,15D  MAX LENGTH OF COMMAND
       LDIR  MOVE IT
********** STEP 5.0 ***************************************
       CALL LOAD+1   CALL NEW "LOAD SUBROUTINE"
********** STEP 6.0 ***************************************
       XOR  A    PASS ZERO IN ACC A TO PHIMON
       CALL STOP  SUBROUTINE TO STOP TAPE DRIVE
********** STEP 7.0 ***************************************
*  FINAL PROCESSING BEFORE RETURN TO PILOT
FINISH      RET
********** STEP 8.0 ***************************************
ERRMSG LD   HL,LOADMS  SET POINTER TO LOAD ERROR MESSAGE
       CALL TV          CALL PHIMON TV EDITOR.
       JP   PHIMON     RETURN TO PHIMON
```

```
*DATA STORAGE AREAS
* FLAG FOR TEST TO SEE IF THIS IS FIRST TIME IN ROUTINE
FIRST  DB,  123
*  ERROR MESSAGES
       DW   240240
LOADMS DB,  377,'PILOT LESSON DESIRED NOT ON THIS TAPE.'
       DB,  'ASK FOR HELP.',015
*********************************************************
*  EQUATES FOR PHIMON REFERENCES
*
*
OVRLAY EQU  341120
LINBUF EQU  346340
LOAD   EQU  347000
STOP   EQU  342015
PHIMON EQU  340042
TV     EQU  343210
*
*  EQUATES FOR PILOT REFERENCES
IPTR   EQU  013AH  INPUT BUFFER POINTER
*********************************************************
ZZZZ   NOP   END MARKER
```

APPENDIX E

ASSEMBLY LANGUAGE LISTING OF ORIGINAL EDITOR

The following pages are the assembly language listing of the original editor [6]. These pages are a reprint of reference 6. The following reprint notice is quoted from reference 6.

"Reprint privileges:

Articles herein that are copyrighted by individual authors or otherwise explicitly marked as having restricted reproduction rights may not be reprinted or copied without permission from Peoples Computer Company, or the authors. All other articles may be reprinted for any non-commercial purpose, provided a credit-line is included. The credit-line should indicate that the material was reprinted from Dr. Dobb's Journal of Computer Calisthenics and Orthodontia, Box E, Menlo Park, CA94025."

The article of reference 6 was not copyrighted.

```
 8080 TEXT EDITOR **
000 100                          * TEXT EDITOR PROGRAM
000 100                          * WRITTEN IN 8080 ASSEMBLY LANGUAGE
000 100                          * VERSION 4  APRIL 16, 1976
000 100                          * TVT OR BAUDOT TTY OUTPUT
000 100                          * USES ROM MONITOR ROUTINES
000 100                          *
000 100                          * SET INITIAL CONDITIONS
000 100                          *
000 100   061 200 347                  LXI  SP,STAK      SET STACK
000 103   076 077                      MVI  A,'?'  SET KILL
000 105   062 043 011                  STA  KILL   CHARACTER
000 110   076 042                      MVI  A,'"'  SET RUBOUT
000 112   062 047 011                  STA  RUBO   CHARACTER
000 115   257                          XRA  A      CLEAR A REGISTER
000 116   062 050 011                  STA  MODE   SET TO LINE MODE
000 121   041 377 035                  LXI  H,1DFFH      SET MAX
000 124   042 051 011                  SHLD MMAX   MEMORY
000 127   315 220 340                  CALL CLRS   CLEAR SCREEN
000 132   041 267 000            IDON LXI  H,EDMS OUTPUT EDIT
000 135   315 102 002                  CALL OUTR   MESSAGE
000 140   076 076                CMRT MVI  A,'>'
000 142   315 071 003                  CALL TVTO   OUTPUT PROMPT
000 145   315 045 004                  CALL DTIN   INPUT COMMAND
000 150   315 145 006                  CALL SCNB   SCAN OFF BLANKS
000 153                          *
000 153                          * INPUT MODE IF COMMAND = C/R
000 153                          *
000 153   332 074 003                  JC   INMD   JUMP TO INPUT MODE
000 156   305                    SCRH PUSH B      SAVE B&C REG
000 157   021 171 007                  LXI  D,CTB4 COMMAND TABLE ADDR
000 162   052 055 011                  LHLD IPNT   COMMAND ADDR IN IBUF
000 165   042 053 011                  SHLD ADDS   FOR SEARCH ROUTINE
000 170   076 004                      MVI  A,4    4 CHAR. COMMANDS
000 172   062 044 011                  STA  NCHR   FOR SEARCH ROUTINE
000 175   006 021                      MVI  B,N4CS # OF 4 CHAR. CMMDS
000 177   315 204 006                  CALL CTSH   CALL SEARCH ROUTINE
000 202                          *
000 202                          * ZERO IF MATCH
000 202                          *
000 202   312 222 000                  JZ   MTCH   JUMP IF MATCH
000 205   076 001                      MVI  A,1    1 CHAR CMMDS
000 207   062 044 011                  STA  NCHR   FOR SEARCH ROUTINE
000 212   006 014                      MVI  B,N1CS # OF 1 CHAR CMMDS
000 214   315 204 006                  CALL CTSH   SEARCH TABLE
000 217   302 020 002                  JNZ  WHAT   ERROR, NO MATCH
000 222   042 067 011            MTCH SHLD TEMP   SAVE ADDRESS
000 225   072 044 011                  LDA  NCHR   # OF CHAR IN CMMD
000 230   117                          MOV  C,A    TO C REG
000 231   006 000                      MVI  B,0    ADD BC TO HL TO
000 233   052 055 011                  LHLD IPNT   GET ADDR OF CHAR
000 236   011                          DAD  B      AFTER CMMD
000 237   042 055 011                  SHLD IPNT   SET IBUF POINTER
000 242   176                          MOV  A,M    GET NEXT CHAR
000 243   376 041                      CPI  ' '+1  TEST IF C/R OR BLANK
000 245   322 020 002                  JNC  WHAT   ERROR IF NOT
000 250   315 145 006            MTGO CALL SCNB   SCAN OFF BLANKS
000 253   042 055 011                  SHLD IPNT   SET IBUF POINTER
000 256   052 067 011                  LHLD TEMP   RECALL MATCH ADDR
000 261   301                          POP  B      RESTORE B&C
000 262   021 140 000            EXCT LXI  D,CMRT SET RETURN ADDRESS
```

```
000 265   325                    PUSH D       TO STACK
000 266   351                    PCHL         EXECUTE CMND
000 267   105 104 111 124   EDMS DW   'EDIT'  EDIT MESSAGE
000 273   215                    DB   8DH     MSB SET ON LAST CHAR
000 274                     *
000 274                     * TAPE OUTPUT COMMAND ROUTINE
000 274                     *
000 274   041 075 001       TPCR LXI H,TRMS   OUTPUT TAB
000 277   315 102 002            CALL OUTR    MESSAGE
000 302   315 045 004            CALL DTIN    INPUT RESPONSE
000 305   053                    DCX  H       SET TO CHAR ADDR
000 306   176                    MOV  A,M     GET RESPONSE
000 307   376 131                CPI  'Y'     TEST IF YES
000 311   312 322 000            JZ   $+6     JUMP IF YES
000 314   376 116                CPI  'N'     TEST IF NO
000 316   302 020 002            JNZ  WHAT    ERROR IF NOT YES OR NO
000 321   257                    XRA  A       CLEAR A REG
000 322   062 045 011            STA  TEM1    SET TAB FLAG
000 325   041 011 010            LXI  H,EFER  OUTPUT FULL OR
000 330   315 102 002            CALL OUTR    PARTIAL MESSAGE
000 333   315 045 004            CALL DTIN    INPUT RESPONSE
000 336   041 072 011            LXI  H,IBUF  SET ADDRESS
000 341   176                    MOV   A,M    GET RESPONSE
000 342   376 106                CPI  'F'     TEST IF FULL
000 344   312 355 000            JZ   STCD    JUMP IF FULL
000 347   376 120                CPI  'P'     TEST IF PARTIAL
000 351   302 020 002            JNZ  WHAT    ERROR IF NOT F OR P
000 354   257                    XRA  A       CLEAR A
000 355   062 067 011       STCD STA  TEMP    SET F OR P MODE
000 360   052 057 011            LHLD TOPL    GET START ADDR
000 363   315 207 341            CALL TMDL    TIME DELAY TO TAPE
000 366   026 001                MVI  D,1     SET TAB COUNTER
000 370   072 045 011       NXTP LDA  TEM1    GET TAB FLAG
000 373   267                    ORA  A       SET 8080 FLAGS
000 374   176                    MOV  A,M     GET DATA FOR OUTPUT
000 375   043                    INX  H       INCREMENT ADDRESS
000 376   312 032 001            JZ   TBOK    JUMP IF TABS STAY
001 001   376 024                CPI  TAB     TEST IF TAB
001 003   302 032 001            JNZ  TBOK    JUMP IF NOT TAB
001 006                     *
001 006                     * CONVERT TABS TO SPACES
001 006                     *
001 006   172                    MOV  A,D     GET COLUMN COUNTER
001 007   326 006                SUI  6       SUBTRACT 6 REPEATEDLY
001 011   322 007 001            JNC  $-5     UNTIL OVERFLOW OCCURS
001 014   137                    MOV  E,A     #(-) OF SPACES TO E REG
001 015   076 040                MVI  A,' '   SPACE
001 017   315 151 341            CALL TAPO    TO TAPE
001 022   024                    INR  D       INCREMENT COLUMN COUNT
001 023   034                    INR  E       INCREMENT SPACE COUNT
001 024   302 017 001            JNZ  $-8     LOOP UNTIL SPACES DONE
001 027   303 370 000            JMP  NXTP    GET NEXT CHAR
001 032   315 151 341       TBOK CALL TAPO    OUTPUT TO TAPE
001 035   024                    INR  D       INCREMENT COLUMN COUNT
001 036   376 001                CPI  1       TEST IF EOF
001 040   312 053 001            JZ   TDON    JUMP IF EOF
001 043   376 015                CPI  13      TEST IF C/R
001 045   302 370 000            JNZ  NXTP    NEXT CHAR IF NOT C/R
001 050   303 366 000            JMP  NXTP-2  RESET COLUMN COUNT
001 053   006 001           TDON MVI  B,1     SET FOR EOF OUTPUT
001 055   072 067 011            LDA  TEMP    GET F/P FLAG
001 060   267                    ORA  A       SET 8080 FLAGS
001 061   302 066 001            JNZ  TPLC    JUMP IF FULL FILE
001 064   004                    INR  B       INCREMENT B TO
001 065   004                    INR  B       EOR
001 066   170               TPLC MOV  A,B     EOF OR EOR TO A
```

```
001 067   315 151 341              CALL TAPO   AND TO TAPE
001 072   303 203 002              JMP  DEOF   DISPLAY END ADDRESS
001 075   122 105 115 117     TRMS DW   'REMOVE TABS?'
001 101   126 105 040 124
001 105   101 102 123 077
001 111   240                      DB   ' '+80H
001 112                       *
001 112                       * NEXT COMMAND ROUTINE
001 112                       *
001 112   001 002 000         NXTC LXI  B,2    SET B&C
001 115   332 243 001              JC   NXUP   UP 1 IF NO NUMBER INPUT
001 120   305                      PUSH B      SAVE B&C
001 121   376 055                  CPI  '-'    TEST IF NEGATIVE
001 123   302 141 001              JNZ  CNVV   JUMP IF POSITIVE
001 126   301                      POP  B      RESTORE B&C
001 127   006 001                  MVI  B,1    SET B FOR SIGN FLAG
001 131   305                      PUSH B      SAVE B&C
001 132   052 055 011              LHLD IPNT   GET NUMBER ADDR
001 135   043                      INX  H      INCREMENT PAST -SIGN
001 136   042 055 011              SHLD IPNT   UPDATE IBUF POINTER
001 141   052 055 011         CNVV LHLD IPNT   GET NUMBER ADDRESS.
001 144   104                      MOV  B,H    SET ADDRESS IN
001 145   115                      MOV  C,L    B&C
001 146   315 323 006              CALL DBCV   CONVERT TO BINARY
001 151   045                      DCR  H      TEST IF LESS THAN 255
001 152   362 020 002              JP   WHAT   JUMP IF HL > 255
001 155   301                      POP  B      RESTORE B&C
001 156   115                      MOV  C,L    # OF LINES TO C
001 157   014                      INR  C      PLUS 1
001 160   170                      MOV  A,B    SIGN FLAG TO A
001 161   267                      ORA  A      SET 8080 FLAGS
001 162   312 243 001              JZ   NXUP   MOVE UP IF POSITIVE
001 165                       *
001 165                       * MOVE BACK TOWARDS TOP
001 165                       *
001 165   175                      MOV  A,L    # OF LINES TO A
001 166   062 067 011         NXBK STA  TEMP   SAVE IN TEMP
001 171   052 057 011              LHLD TOPL   FILE START ADDR
001 174   353                      XCHG        TO DE REGISTERS
001 175   052 061 011              LHLD PNTR   CURRENT ADDRESS
001 200   053                 NATP DCX  H      BACK UP 2
001 201   053                      DCX  H      CHARACTERS
001 202   315 327 003              CALL OVTS   TEST IF AT TOP
001 205   332 217 001              JC   NAOK   JUMP IF NOT TOP
001 210   353                      XCHG        TOP ADDR TO HL
001 211   042 061 011              SHLD PNTR   SET LINE POINTER
001 214   303 263 001              JMP  NDON   JUMP TO EXIT
001 217                       *
001 217                       * BACK UP TO START OF CURRENT LINE
001 217                       *
001 217   176                 NAOK MOV  A,M    GET CURRENT CHAR.
001 220   376 015                  CPI  13     TEST FOR C/R
001 222   302 201 001              JNZ  NATP+1 LOOP UNTIL C/R FOUND
001 225   043                      INX  H      SET ADDR TO LINE START
001 226   042 061 011              SHLD PNTR   SET LINE POINTER
001 231   072 067 011              LDA  TEMP   # OF LINES TO A
001 234   075                      DCR  A      SUBTRACT 1
001 235   302 166 001              JNZ  NXBK   LOOP UNTIL # = ZERO
001 240   303 263 001              JMP  NDON   JUMP TO EXIT
001 243                       *
001 243                       * MOVE TOWARDS END OF FILE
001 243                       *
001 243   015                 NXUP DCR  C      # OF LINES - 1
001 244   312 263 001              JZ   NDON   JUMP IF DONE
001 247   315 003 002              CALL NLST   GET NEXT LINE ADDR
001 252   332 336 005              JC   BOTM   JUMP IF AT EOF
```

```
001 255   042 061 011        SHLD PNTR   SET LINE POINTER
001 260   303 243 001        JMP  NXUP   LOOP UNTIL DONE
001 263                 *
001 263                 * EXIT TO PRINT ROUTINE
001 263                 *
001 263   072 050 011   NDON LDA  MODE   MODE FLAG TO A
001 266   052 061 011        LHLD PNTR   LOAD LINE POINTER
001 271   267                ORA  A      SET 8080 FLAGS
001 272   302 105 006        JNZ  LIST   JUMP TO PAGE OUTPUT
001 275   315 220 340        CALL CLRS   CLEAR SCREEN
001 300   303 016 003        JMP  LNOT   JUMP TO LINE OUTPUT
001 303                 *
001 303                 * RUBOUT COMMAND ROUTINE
001 303                 * SETS NEW RUBOUT CHARACTER
001 303                 *
001 303   052 055 011   RBCM LHLD IPNT   LOAD IBUF POINTER
001 306   176                MOV  A,M    GET NEW RUBOUT CHAR
001 307   376 041            CPI  ' '+1  TEST IF BLANK OR
001 311   332 020 002        JC   WHAT   CONTROL CHAR
001 314   062 047 011        STA  RUBO   VALID, SET RUBOUT
001 317   311                RET         EXIT TO EDIT MONITOR
001 320                 *
001 320                 * OUTPUT MODE SET COMMAND
001 320                 *
001 320   052 055 011   MSCR LHLD IPNT   LOAD IBUF ADDRESS
001 323   176                MOV  A,M    GET NEW MODE
001 324   376 120            CPI  'P'    TEST FOR PAGE
001 326   312 337 001        JZ   STMD   JUMP IF PAGE MODE
001 331   376 114            CPI  'L'    TEST IF LINE
001 333   302 020 002        JNZ  WHAT   ERROR IF NOT P OR N
001 336   257                XRA  A      CLEAR A
001 337   062 050 011   STMD STA  MODE   SET MODE
001 342   311                RET         DONE
001 343                 *
001 343                 * TOP COMMAND ROUTINE
001 343                 *
001 343   052 057 011   TOPR LHLD TOPL   LOAD TOP ADDRESS
001 346   042 061 011        SHLD PNTR   SET LINE POINTER
001 351                 *
001 351                 * JUMP TO LINE OR PAGE OUTPUT
001 351                 * AS DETERMINED BY CURRENT MODE
001 351                 *
001 351   072 050 011        LDA  MODE
001 354   267                ORA  A
001 355   302 105 006        JNZ  LIST   PAGE MODE
001 360   315 220 340   TPLE CALL CLRS
001 363   303 016 003        JMP  LNOT   LINE MODE
001 366                 *
001 366                 * KILL COMMAND ROUTINE
001 366                 *
001 366   052 055 011   KLRT LHLD IPNT   LOAD IBUF ADDRESS
001 371   176                MOV  A,M    GET NEW KILL CHAR
001 372   376 041            CPI  ' '+1  STORE NEW KILL
001 374   332 020 002        JC   WHAT   UNLESS SPACE OR
001 377   062 043 011        STA  KILL   CONTROL CHARACTER
002 002   311                RET
002 003                 *
002 003                 * NLST GETS NEXT LINE ADDRESS
002 003                 *
002 003   052 061 011   NLST LHLD PNTR   GET CURRENT ADDRESS
002 006   176                MOV  A,M    GET CURRENT CHAR
002 007   043                INX  H      INCR. ADDR
002 010   376 015            CPI  13     TEST FOR C/R
002 012   310                RZ          DONE IF C/R
002 013   322 006 002        JNC  NLST+3 LOOP IF NOT EOF
002 016   053                DCX  H      BACK UP 1
```

```
002 017   311                       RET
002 020                        *
002 020                        * ERROR MESSAGE OUTPUT
002 020                        *
002 020   041 034 002          WHAT LXI  H,WTMS LOAD MESSAGE ADDR
002 023   061 200 347               LXI  SP,STAK       RESET STACK
002 026   315 102 002               CALL OUTR   OUTPUT MESSAGE
002 031   303 140 000               JMP  CMRT   EXIT TO EDIT MONITOR
002 034   040 127 110 101      WTMS DW   ' WHAT?'
002 040   124 077
002 042   215                       DB   8DH
002 043                        *
002 043                        * NEW FILE COMMAND
002 043                        *
002 043   041 075 002          NEWF LXI  H,ADMS LOAD ADDRESS MSG ADDR
002 046   315 102 002               CALL OUTR   OUTPUT MESSAGE
002 051   315 236 006               CALL HLIN   INPUT ADDRESS
002 054   042 061 011               SHLD PNTR   SET LINE POINTER
002 057   042 063 011               SHLD EFPN   SET EOF POINTER
002 062   042 057 011               SHLD TOPL   SET TOP POINTER
002 065   066 001                   MVI  M,1    PUT EOF IN FILE AREA
002 067   315 076 340               CALL CRLF   OUTPUT C/R
002 072   303 074 003               JMP  INMD   JUMP TO INPUT ROUTINE
002 075   101 104 104 122      ADMS DW   'ADDR'
002 101   240                       DB   ' '+80H
002 102                        *
002 102                        * MESSAGE OUTPUT ROUTINE
002 102                        *
002 102   176                  OUTR MOV  A,M    LOAD CHARACTER
002 103   006 002                   MVI  B,2    B = 2
002 105   267                       ORA  A      SET FLAGS
002 106   362 114 002               JP   OTCH   JUMP IF MSB ZERO
002 111   326 200                   SUI  80H    CLEAR MSB
002 113   005                       DCR  B      B = 1 NOW
002 114   315 071 003          OTCH CALL TVTO   OUTPUT CHARACTER
002 117   005                       DCR  B      B = 0 OR 1 NOW
002 120   310                       RZ          DONE IF B = 0
002 121   043                       INX  H      INCREMENT ADDRESS
002 122   303 102 002               JMP  OUTR   LOOP UNTIL DONE
002 125                        *
002 125                        * DISPLAY  POINTER
002 125                        *
002 125   052 061 011          DCPL LHLD PNTR   LOAD POINTER
002 130   353                       XCHG        TO D&E
002 131   001 156 002               LXI  B,OTMS LOAD MESSAGE ADDR
002 134   041 173 002               LXI  H,OTLC OCTAL STORAGE ADDR
002 137   173                  EEOF MOV  A,E    LOW BYTE TO A
002 140   315 064 007               CALL BINH+1 CONVERT TO OCTAL
002 143   172                       MOV  A,D    HIGH BYTE TO A
002 144   043                       INX  H      INCREMENT STORAGE ADDR
002 145   315 063 007               CALL BINH   CONVERT HIGH BYTE
002 150   140                       MOV  H,B    MESSAGE ADDR
002 151   151                       MOV  L,C    TO HL REGISTERS
002 152   315 102 002               CALL OUTR   OUTPUT RESULTS
002 155   311                       RET         EXIT
002 156   120 117 111 116      OTMS DW   'POINTER (LH) '
002 162   124 105 122 040
002 166   050 114 110 051
002 172   040
002 173                        OTLC DS   3
002 176   040                       DB   ' '
002 177                             DS   3
002 202   215                       DB   6DH
002 203                        *
002 203                        * DISPLAY EOF LOCATION
002 203                        *
```

```
002 203   052 063 011         DEOF LHLD EFPN   LOAD EOF ADDRESS
002 206   353                      XCHG        TO DE REGISTERS
002 207   001 220 002              LXI  B,EMSG LOAD MESSAGE ADDR
002 212   041 231 002              LXI  H,EOLC LOAD STORAGE ADDR
002 215   303 137 002              JMP  EEOF   JUMP TO DISPLAY
002 220   105 117 106 040     EMSG DW   'EOF (LH) '
002 224   050 114 110 051
002 230   040
002 231                       EOLC DS   3
002 234   040                      DB   ' '
002 235                            DS   3
002 240   215                      DB   8DH
002 241                       *
002 241                       * DISPLAY  MAX  MEMORY  VALUE
002 241                       *
002 241   052 051 011         DISM LHLD MMAX   LOAD MAX MEM ADDR
002 244   353                      XCHG        TO DE REGISTERS
002 245   001 256 002              LXI  B,MXSG LOAD MSSG ADDR
002 250   041 273 002              LXI  H,MXLC LOAD STORAGE ADDR
002 253   303 137 002              JMP  EEOF   JUMP TO DISPLAY
002 256   115 101 130 040     MXSG DW   'MAX MEM (LH) '
002 262   115 105 115 040
002 266   050 114 110 051
002 272   040
002 273                       MXLC DS   3
002 276   040                      DB   ' '
002 277                            DS   3
002 302   215                      DB   8DH
002 303                       *
002 303                       * TAPE INPUT COMMAND
002 303                       *
002 303   041 075 002         ITCR LXI  H,ADMS OUTPUT ADDR
002 306   315 102 002              CALL OUTR   MESSAGE
002 311   315 236 006              CALL HLIN   INPUT ADDRESS
002 314   042 061 011              SHLD PNTR   SET LINE POINTER
002 317   042 057 011              SHLD TOPL   SET TOP POINTER
002 322   315 045 004              CALL DTIN   WAIT FOR C/R
002 325   052 057 011              LHLD TOPL   LOAD TOP ADDR
002 330   315 263 340         TPIN CALL TAPI   GET DATA FROM TAPE
002 333   162                      MOV  M,D    MOVE DATA TO MEMORY
002 334   376 001                  CPI  1      TEST FOR EOF
002 336   312 345 002              JZ   TDIN   JUMP IF EOF
002 341   043                      INX  H      INCREMENT ADDRESS
002 342   303 330 002              JMP  TPIN   LOOP
002 345   042 063 011         TDIN SHLD EFPN   SET EOF POINTER
002 350   315 302 003              CALL EFFN   TEST FOR OVERFLOW BY
002 353   311                      RET         SEARCHING FOR EOF
002 354                       *
002 354                       * EDIT COMMAND ROUTINE
002 354                       *
002 354   041 075 002         EDCR LXI  H,ADMS OUTPUT ADDR
002 357   315 102 002              CALL OUTR   MESSAGE
002 362   315 236 006              CALL HLIN   INPUT ADDRESS
002 365   042 061 011              SHLD PNTR   SET LINE POINTER
002 370   042 057 011              SHLD TOPL   SET TOP POINTER
002 373   315 302 003              CALL EFFN   FIND  EOF
002 376   042 063 011              SHLD EFPN   SET EOF POINTER
003 001   052 061 011              LHLD PNTR   LOAD CURRENT ADDR
003 004   072 050 011              LDA  MODE   LOAD MODE FLAG
003 007   267                      ORA  A      SET 8080 FLAGS
003 010   302 105 006              JNZ  LIST   JUMP IF PAGE MODE
003 013   315 220 340              CALL CLRS   CLEAR SCREEN
003 016                       *
003 016                       * LINE OUTPUT ROUTINE.
003 016                       *
003 016   006 001             LNOT MVI  B,1    SET COLUMN COUNT
```

```
003 020    176                     MOV  A,M    GET CHARACTER
003 021    376 002                 CPI  2      TEST FOR EOF
003 023    330                     RC          RETURN IF EOF
003 024    043                     INX  H      INCREMENT ADDRESS
003 025    376 024                 CPI  TAB    TEST FOR TAB
003 027    302 040 003             JNZ  LNO1   JUMP IF NOT TAB
003 032    315 362 006             CALL TBST   CONVERT TAB TO SPACES
003 035    303 020 003             JMP  LNOT+2 LOOP
003 040    004                LNO1 INR  B      INCREMENT COLUMN COUNT
003 041    315 052 003             CALL DTOT   OUTPUT CHARACTER
003 044    376 015                 CPI  13     TEST IF C/R
003 046    310                     RZ          RETURN IF C/R
003 047    303 020 003             JMP  LNOT+2 LOOP
003 052                       *
003 052                       * DATA OUT ROUTINE.  CHANGED BETWEEN
003 052                       * TVT AND TTY OUTPUT DURING PROGRAM
003 052                       * EXECUTION.  ALL REGISTERS PRESERVED
003 052                       *
003 052    345                DTOT PUSH H      SAVE ALL REGISTERS
003 053    325                     PUSH D
003 054    305                     PUSH B
003 055    365                     PUSH PSW
003 056                       *
003 056                       * FOLLOWING INSTRUCTION CHANGED
003 056                       * BY OUTM COMMAND (TVT OR TTY OUT)
003 056                       *
003 056    315 071 003        DTO1 CALL TVTO   OUTPUT
003 061    361                     POP  PSW    RESTORE REGISTERS
003 062    301                     POP  B
003 063    321                     POP  D
003 064    341                     POP  H
003 065    311                     RET
003 066                       *
003 066                       * TVT INPUT ROUTINE
003 066                       *
003 066    333 000            TVTI IN   TVT    INPUT DATA
003 070    311                     RET
003 071                       *
003 071                       * TVT OUTPUT ROUTINE
003 071                       *
003 071    323 000            TVTO OUT  TVT    OUTPUT DATA
003 073    311                     RET
003 074                       *
003 074                       * INPUT COMMAND ROUTINE
003 074                       *
003 074    061 200 347        INMD LXI  SP,STAK      RESET STACK
003 077    041 003 010             LXI  H,INMS MSSG ADDR
003 102    315 102 002             CALL OUTR   OUTPUT MESSAGE
003 105    315 045 004        INLP CALL DTIN   INPUT NEW LINE
003 110    041 072 011             LXI  H,IBUF LOAD IBUF START ADDR
003 113    176                     MOV  A,M    GET FIRST CHARACTER
003 116    376 015                 CPI  13     TEST FOR C/R
003 116    312 132 000             JZ   IDON   DONE IF C/R
003 121    315 003 002             CALL NLST   GET NEXT LINE ADDR
003 124    042 061 011             SHLD PNTR   SET LINE POINTER
003 127    042 065 011             SHND MVAD   SET MOVE LIMIT
003 132    315 140 003             CALL CENT   INSERT LINE
003 135    303 105 003             JMP  INLP   LOOP FOR ANOTHER INPUT
003 140                       *
003 140                       * LINE INSERT ROUTINE
003 140                       *
003 140    052 063 011        CENT LHLD EFPN   LOAD EOF ADDR
003 143    014                     INR  C      INCREMENT CHAR COUNT
003 144    006 000                 MVI  B,0    B = 0
003 146    011                     DAD  B      EOF ADDR + CHAR COUNT
003 147    353                     XCHG        IS NEW EOF (TO DE)
```

```
003 150    052 051 011          LHLD MMAX   LOAD MAX MEM VALUE
003 153    315 327 003          CALL OVTS   TEST FOR OVERFLOW
003 156    322 173 004          JNC  MOFL   JUMP IF OVERFLOW
003 161    052 063 011     MSOK LHLD EFPN   LOAD EOF ADDR
003 164    315 221 003          CALL RMOV   MOVE FILE UP
003 167    052 063 011          LHLD EFPN   LOAD EOF ADDR
003 172    011                  DAD  B      ADD CHAR. COUNT
003 173    042 063 011          SHLD EFPN   SET NEW EOF ADDR
003 176    052 055 011          LHLD IPNT   NEW LINE ADDR (IBUF)
003 201    015                  DCR  C      CHAR COUNT - 1
003 202    011                  DAD  B      FORM LINE END ADDR
003 203    042 065 011          SHLD MVAD   SET MOVE LIMIT
003 206    052 055 011          LHLD IPNT   LINE START ADDR
003 211    353                  XCHG        TO DE REGISTERS
003 212    052 061 011          LHLD PNTR   LOAD INSERT START ADDR
003 215    315 012 004          CALL LMOV   MOVE IN NEW LINE
003 220    311                  RET
003 221                    *
003 221                    * RMOV - RIGHT (UP) MOVE.  MOVES DATA
003 221                    * FROM HL ADDRESS TO DE ADDRESS UNTIL
003 221                    * HL IS DECREMENTED TO MVAD
003 221                    * ADDRESS (INCLUSIVE)
003 221                    *
003 221    305             RMOV PUSH B      SAVE B&C
003 222    104                  MOV  B,H    SOURCE ADDR TO
003 223    115                  MOV  C,L    BC REGISTERS
003 224    052 065 011          LHLD MVAD   LOAD LIMIT ADDR
003 227    012             NXRM LDAX B      GET DATA
003 230    022                  STAX D      STORE AT NEW ADDR
003 231    175                  MOV  A,L    TEST IF AT
003 232    271                  CMP  C      LOW LIMIT
003 233    302 243 003          JNZ  RMCT   JUMP IF NOT
003 236    174                  MOV  A,H    TEST IF AT
003 237    270                  CMP  B      HIGH LIMIT
003 240    312 250 003          JZ   RDON   JUMP IF AT LIMIT
003 243    013             RMCT DCX  B      DECREMENT
003 244    033                  DCX  D      ADDRESSES
003 245    303 227 003          JMP  NXRM   MOVE NEXT CHAR
003 250    301             RDON POP  B      RESTORE B&C
003 251    311                  RET
003 252                    *
003 252                    * SINGLE LINE INPUT COMMAND
003 252                    *
003 252    332 074 003     INSL JC   INMD   JUMP IF NO STRING
003 255    015                  DCR  C      DECREMENT CHAR COUNT
003 256    015                  DCR  C      TWICE
003 257    041 074 011          LXI  H,IBUF+2     LINE START ADDR
003 262    042 055 011          SHLD IPNT   SET IBUF POINTER
003 265    315 003 002          CALL NLST   GET NEXT LINE ADDR
003 270    042 065 011          SHLD MVAD   SET MOVE LIMIT
003 273    042 061 011          SHLD PNTR   SET LINE POINTER
003 276    315 140 003          CALL CENT   INSERT LINE
003 301    311                  RET
003 302                    *
003 302                    * EFFN ROUTINE - FINDS EOF AND
003 302                    * TESTS FOR MEMORY OVERFLOW
003 302                    *
003 302    052 051 011     EFFN LHLD MMAX   LOAD MAX MEM VALUE
003 305    353                  XCHG        TO DE REGISTERS
003 306    052 061 011          LHLD PNTR   LOAD CURRENT ADDR
003 311    315 327 003     EFF1 CALL OVTS   TEST FOR OVERFLOW
003 314    332 173 004          JC   MOFL   JUMP IF OVERFLOW
003 317    176                  MOV  A,M    GET CHARACTER
003 320    376 001              CPI  1      TEST FOR EOF
003 322    310                  RZ          RETURN IF EOF
003 323    043                  INX  H      INCREMENT ADDRESS
```

```
003 324   303 311 003         JMP  EFF1   LOOP
003 327                  *
003 327                  * HL/DE COMPARE - CARRY SET IF
003 327                  * HL GREATER THAN DE
003 327                  *
003 327   173            OVTS MOV  A,E
003 330   225                 SUB  L      FORM E - L
003 331   172                 MOV  A,D
003 332   234                 SBB  H      FORM D-H-BORROW
003 333   311                 RET
003 334                  *
003 336                  * DELETE COMMAND ROUTINE
003 334                  *
003 334   016 002        DELE MVI  C,2    # OF LINES + 1
003 336   332 357 003         JC   DLOC   JUMP IF NO # INPUT
003 341   052 055 011         LHLD IPNT   LOAD # ADDR
003 344   104                 MOV  B,H    TO B&C REGISTERS
003 345   115                 MOV  C,L
003 346   315 325 006         CALL DBCV   CONVERT # TO BINARY
003 351   115                 MOV  C,L    RESULT TO C
003 352   014                 INR  C      PLUS 1
003 353   045                 DCR  H      TEST IF < 256
003 354   362 020 002         JP   WHAT   ERROR IF > 255
003 357   015            DLOC DCR  C      DECREMENT LINE COUNT
003 360   310                 RZ          DONE IF ZERO
003 361   052 063 011         LHLD EFPN   LOAD EOF ADDR
003 364   042 065 011         SHLD MVAD   SET MOVE LIMIT
003 367                  *
003 367                  * GET NEXT LINE START ADDRESS AS
003 367                  * MOVE START ADDRESS
003 367                  *
003 367   315 003 002         CALL NLST
003 372   332 336 005         JC   BOTM   JUMP IF AT EOF
003 375   353                 XCHG        SOURCE ADDR TO DE
003 376   052 061 011         LHLD PNTR   DESTINATION ADDR
004 001   315 012 004         CALL LMOV   MOVE FILE DOWN
004 004   042 063 011         SHLD EFPN   SET NEW EOF ADDR
004 007   303 357 003         JMP  DLOC   LOOP
004 012                  *
004 012                  * LEFT (DOWN) MOVE.  MOVES DATA
004 012                  * FROM DE ADDR TO HL ADDR UNTIL
004 012                  * DE INCREMENTED TO MVAD LIMIT
004 012                  *
004 012   305            LMOV PUSH B      SAVE B&C
004 013   102                 MOV  B,D    SOURCE ADDR TO BC
004 014   113                 MOV  C,E
004 015   353                 XCHG        SAVE HL IN DE
004 016   052 065 011         LHLD MVAD   LOAD LIMIT ADDR
004 021   353                 XCHG        ADDR'S TO PROPER REGS
004 022   012            LMLP LDAX B      GET CHARACTER
004 023   167                 MOV  M,A    STORE AT NEW ADDR
004 024   171                 MOV  A,C    TEST LOW LIMIT
004 025   273                 CMP  E
004 026   302 036 004         JNZ  LMCT   JUMP IF NOT AT LIMIT
004 031   170                 MOV  A,B    TEST HIGH LIMIT
004 032   272                 CMP  D
004 033   312 043 004         JZ   LDON   JUMP IF AT LIMIT
004 036   043            LMCT INX  H      INCREMENT ADDRESSES
004 037   003                 INX  B
004 040   303 022 004         JMP  LMLP   LOOP
004 043   301            LDON POP  B      RESTORE B&C
004 044   311                 RET
004 045                  *
004 045                  * DATA INPUT ROUTINE
004 045                  *
004 045   041 072 011    DTIN LXI  H,IBUF LOAD BUFFER ADDR
```

```
004 050   042 055 011             SHLD IPNT   SET BUFFER POINTER
004 053   072 043 011             LDA  KILL   KILL CHARACTER
004 056   137                     MOV  E,A    TO E REGISTER
004 057   072 047 011             LDA  RUBO   RUBOUT CHARACTER
004 062   127                     MOV  D,A    TO D REGISTER
004 063   001 000 001             LXI  B,100H SET B&C
004 066   315 066 003        NXCH CALL TVTI   INPUT FROM TVT
004 071   273                     CMP  E      TEST IF KILL
004 072   312 045 004             JZ   DTIN   JUMP IF KILL
004 075   272                     CMP  D      TEST IF RUBOUT
004 076   302 112 004             JNZ  STOR   JUMP IF NOT RUBOUT
004 101   015                     DCR  C      DECREMENT CHAR COUNT
004 102   372 045 004             JM   DTIN   JUMP IF NEGATIVE
004 105   005                     DCR  B      DECREMENT COLUMN COUNT
004 106   053                     DCX  H      DECREMENT ADDR
004 107   303 066 004             JMP  NXCH   GET NEXT CHARACTER
004 112   167                STOR MOV  M,A    CHARACTER TO BUFFER
004 113   376 015                 CPI  13     TEST FOR C/R
004 115   310                     RZ          RETURN IF C/R
004 116   171                     MOV  A,C    CHAR COUNT TO A REG
004 117   376 110                 CPI  CMAX   TEST FOR MAX INPUT
004 121   312 066 004             JZ   NXCH   LOOP IF MAX
004 124   176                     MOV  A,M    RECALL CHAR
004 125   043                     INX  H      INCREMENT ADDRESS
004 126   014                     INR  C      INCREMENT CHAR COUNT
004 127   004                     INR  B      INCREMENT COLUMN COUNT
004 130   376 024                 CPI  TAB    TEST FOR TAB
004 132   302 066 004             JNZ  NXCH   NEXT CHAR IF NOT TAB
004 135   005                     DCR  B      DECREMENT COLUMN COUNT
004 136   315 362 006             CALL TBST   ECHO BACK SPACES
004 141   303 066 004             JMP  NXCH   GET NEXT CHAR
004 144                      *
004 144                      * SET MAX MEMORY COMMAND
004 144                      *
004 144   041 164 004        MMCR LXI  H,MADS OUTPUT MAX
004 147   315 102 002             CALL OUTR   MEM MESSAGE
004 152   315 236 006             CALL HLIN   INPUT ADDRESS
004 155   042 051 011             SHLD MMAX   SET MAX MEM
004 160   315 076 340             CALL CRLF   C/R OUTPUT TO TVT
004 163   311                     RET
004 164   115 101 130 040    MADS DW   'MAX ME'
004 170   115 105
004 172   315                     DB   'M'+80H
004 173                      *
004 173                      * MEMORY OVERFLOW ERROR
004 173                      *
004 173   041 212 004        MOFL LXI  H,OFMS MESSAGE ADDR
004 176   061 200 347             LXI  SP,STAK      RESET STACK
004 201   315 076 340             CALL CRLF   C/R TO TVT
004 204   315 102 002             CALL OUTR   OUTPUT MESSAGE
004 207   303 140 000             JMP  CMRT   RETURN
004 212   115 105 115 040    OFMS DW   'MEM OVERFLOW'
004 216   117 126 105 122
004 222   106 114 117 127
004 226   215                     DB   8DH
004 227                      *
004 227                      * APPEND COMMAND
004 227                      *
004 227   171                APND MOV  A,C    CHAR COUNT TO A
004 230   326 003                 SUI  3      MINUS 3
004 232   332 020 002             JC   WHAT   ERROR IF ONLY 2 INPUT
004 235   015                     DCR  C      DECREMENT CHAR COUNT
004 236   015                     DCR  C      TWICE
004 237   315 003 002             CALL NLST   NEXT LINE ADDR
004 242   332 336 005             JC   BOTM   JUMP IF AT EOF
004 245   053                     DCX  H      SET ADDR TO C/R
```

```
004 246   042 067 011                SHLD TEMP   SET TEMP FOR CHANGE
004 251   041 073 011                LXI  H,IBUF+1     STRING ADDR - 1
004 254   042 045 011                SHLD TEM1   SET TEM1 FOR CHANGE
004 257   006 000                    MVI  B,0    B = 0
004 261   303 007 005                JMP  CNGO   JUMP TO CHANGE ROUTINE
004 264                         *
004 264                         * CHANGE COMMAND
004 264                         *
004 264   332 020 002           CCRT JC   WHAT   ERROR IF NO STRING
004 267   052 055 011                LHLD IPNT   DELIMITER ADDR
004 272   126                        MOV  D,M    FIRST DELIMITER
004 273   006 000                    MVI  B,0    ZERO CHAR COUNTER
004 275                         *
004 275                         * COUNT CHARACTERS IN FIRST STRING
004 275                         *
004 275   043                   CCR1 INX  H      CHAR ADDRESS
004 276   176                        MOV  A,M    LOAD CHARACTER
004 277   376 015                    CPI  13     TEST FOR C/R
004 301   312 020 002                JZ   WHAT   ERROR IF C/R
004 304   272                        CMP  D      TEST FOR DELIMITER
004 305   312 314 004                JZ   LSTR   JUMP IF DELIMITER
004 310   004                        INR  B      INCREMENT CHAR COUNT
004 311   303 275 004                JMP  CCR1   LOOP
004 314   042 045 011           LSTR SHLD TEM1   SAVE DELIMITER ADDR
004 317   052 055 011                LHLD IPNT   FIRST DELIMITER ADDR
004 322   043                        INX  H      INCREMENT TO STRING
004 323   042 055 011                SHLD IPNT   ADDR AND SAVE
004 326   052 061 011                LHLD PNTR   LOAD CURRENT FILE ADDR
004 331   110                   STSH MOV  C,B    CHAR COUNT TO C
004 332   042 067 011                SHLD TEMP   SAVE CURRENT ADDR
004 335   353                        XCHG        ADDR TO DE
004 336   052 055 011                LHLD IPNT   STRING 1 ADDR
004 341   315 163 006                CALL SEAR   SEARCH FOR STRING
004 344   312 367 004                JZ   STMT   JUMP IF FOUND
004 347   052 067 011                LHLD TEMP   FILE ADDR
004 352   176                        MOV  A,M    CHARACTER FROM LINE
004 353   376 015                    CPI  13     TEST FOR C/R
004 355   312 165 005                JZ   CCDN   JUMP TO EXIT IF C/R
004 360   332 336 005                JC   BOTM   AT BOTTOM IF EOF
004 363   043                        INX  H      INCREMENT ADDR TO NEXT
004 364   303 331 004                JMP  STSH   CHAR AND CONTINUE SEARCH
004 367                         *
004 367                         * COUNT CHARACTERS IN SECOND STRING
004 367                         *
004 367   052 045 011           STMT LHLD TEM1   DELIMITER ADDR
094 372   016 000                    MVI  C,0    ZERO CHAR COUNTER
004 374   043                   STM1 INX  H      INCREMENT ADDR
004 375   176                        MOV  A,M    LOAD STRING 2 CHAR
004 376   376 015                    CPI  13     TEST FOR C/R
005 000   312 007 005                JZ   CNGO   END OF STRING 2 IF C/R
005 003   014                        INR  C      INCREMENT CHAR COUNT
005 004   303 374 004                JMP  STM1   LOOP UNTIL C/R
005 007                         *
005 007                         * BEGIN CHANGE.  B CONTAINS # OF CHAR
005 007                         * IN STRING 1; C HAS # IN STRING 2
005 007                         *
005 007   170                   CNGO MOV  A,B    B TO A
005 010   271                        CMP  C      COMPARE STRING 2 LENGTH
005 011   312 142 005                JZ   EQUL   JUMP IF EQUAL
005 014   322 100 005                JNC  LESS   JUMP IF B > C
005 017                         *
005 017                         * HERE IF FILE LENGTH INCREASES
005 017                         *
005 017   052 067 011                LHLD TEMP   LOAD MATCH ADDR
005 022   205                        ADD  L      ADD STRING 1 LENGTH
005 023   157                        MOV  L,A    TO FORM ADDRESS
```

```
005 024   174                  MOV  A,H    WHERE INSERT MODE STARTS
005 025   316 000              ACI  0      ADD CARRY
005 027   147                  MOV  H,A
005 030   171                  MOV  A,C    NUMBER TO ADD IS STRING 2
005 031   220                  SUB  B      - STRING 1 (LENGTHS)
005 032   042 065 011          SHLD MVAD   SET MOVE LIMIT
005 035   052 063 011          LHLD EFPN   LOAD EOF ADDR
005 040   305                  PUSH B      SAVE STRING LENGTHS
005 041   117                  MOV  C,A    DIFFERENCE TO C
005 042   006 000              MVI  B,0    EOF PLUS DIFFERENCE
005 044   011                  DAD  B      IS NEW EOF ADDR
005 045   353                  XCHG        NEW EOF ADDR TO DE
005 046   052 051 011          LHLD MMAX   MAX MEM VALUE
005 051   315 327 003          CALL OVTS   TEST FOR OVERFLOW
005 054   322 173 004          JNC  MOFL   JUMP IF OVERFLOW
005 057   052 063 011          LHLD EFPN   LOAD EOF ADDR
005 062   315 221 003          CALL RMOV   MOVE FILE UP
005 065   052 063 011          LHLD EFPN   LOAD EOF ADDR
005 070   011                  DAD  B      FORM NEW EOF ADDR
005 071   042 063 011          SHLD EFPN   SAVE NEW EOF ADDR
005 074   301                  POP  B      RESTORE STRING LENGTHS
005 075   303 142 005          JMP  EQUL   INSERT NEW STRING
005 100                   *
005 100                   * HERE IF FILE SIZE DECREASES
005 100                   *
005 100   052 067 011     LESS LHLD TEMP   LOAD MATCH ADDR
005 103   175                  MOV  A,L    LOW ADDR TO A
005 104   201                  ADD  C      ADD STRING 2 LENGTH
005 105   157                  MOV  L,A    TO FORM MOVE
005 106   174                  MOV  A,H    DESTINATION ADDR
005 107   316 000              ACI  0      ADD CARRY
005 111   147                  MOV  H,A
005 112   353                  XCHG        SAVE IN DE
005 113   052 063 011          LHLD EFPN   LOAD EOF ADDR
005 116   042 065 011          SHLD MVAD   SET MOVE LIMIT
005 121   052 067 011          LHLD TEMP   LOAD MATCH ADDR
005 124   175                  MOV  A,L    FORM MOVE START
005 125   200                  ADD  B      AS MATCH ADDR PLUS
005 126   157                  MOV  L,A    STRING 1 LENGTH
005 127   174                  MOV  A,H
005 130   316 000              ACI  0      ADD CARRY TO HIGH ADDR
005 132   147                  MOV  H,A
005 133   353                  XCHG        DE=SOURCE, HL=DEST
005 134   315 012 004          CALL LMOV   MOVE FILE
005 137   042 063 011          SHLD EFPN   SET NEW EOF ADDR
005 142                   *
005 142                   * HERE IF FILE SIZE UNCHANGED
005 142                   *
005 142   052 067 011     EQUL LHLD TEMP   MATCH ADDR
005 145   353                  XCHG        TO DE
005 146   052 045 011          LHLD TEM1   SECOND DELIMITER ADDR
005 151   043                  INX  H      TO STRING 2 ADDR
005 152   015             EQLP DCR  C      DECREMENT STRING 2 COUNT
005 153   372 165 005          JM   CCDN   JUMP IF NEGATIVE
005 156   176                  MOV  A,M    GET STRING 2 CHAR
005 157   022                  STAX D      PUT IN FILE
005 160   043                  INX  H      INCREMENT ADDRESSES
005 161   023                  INX  D
005 162   303 152 005          JMP  EQLP   LOOP
005 165   052 061 011     CCDN LHLD PNTR   LINE START ADDR
005 170   072 050 011          LDA  MODE   OUTPUT MODE FLAG
005 173   267                  ORA  A      SET 8080 FLAGS
005 174   312 016 003          JZ   LNOT   LINE MODE OUTPUT
005 177   303 105 006          JMP  LIST   PAGE MODE OUTPUT
005 202                   *
005 202                   * LOC1 ROUTINE  - FINDS LENGTH OF STRING
```

```
005 202                         * STARTING IN COLUMN 2 OF IBUF
005 202                         *
005 202   041 073 011           LOC1 LXI  H,IBUF+1      1ST LOCATION
005 205   006 376                    MVI  B,-2    CHAR COUNTER
005 207   176                   LNCX MOV  A,M     GET CHARACTER
005 210   376 015                    CPI  13      TEST FOR C/R
005 212   312 222 005                JZ   LOC2    JUMP IF C/R
005 215   004                        INR  B       INCREMENT CHAR COUNT
005 216   043                        INX  H       INCREMENT ADDR
005 217   303 207 005                JMP  LNCX    LOOP
005 222   170                   LOC2 MOV  A,B     COUNTER TO A
005 223   267                        ORA  A       SET FLAGS
005 224   372 020 002                JM   WHAT    ERROR IF STRING < 2
005 227   004                        INR  B       SET TO ACTUAL COUNT
005 230   311                        RET
005 231                         *
005 231                         * LOCATE COMMAND - STRING SEARCH
005 231                         * AT EACH CHARACTER POSITION.  FROM
005 231                         * NEXT LINE START TO EOF
005 231                         *
005 231   315 202 005           LOCT CALL LOC1    GET STRING LENGTH
005 234   041 074 011                LXI  H,IBUF+2      STRING ADDR
005 237   042 055 011                SHLD IPNT    SET IBUF POINTER
005 242   315 003 002                CALL NLST    NEXT LINE ADDR
005 245   332 336 005                JC   BOTM    JUMP IF AT EOF
005 250   110                   LNCH MOV  C,B     STRING LENGTH TO C
005 251   042 067 011                SHLD TEMP    SAVE FILE ADDR
005 254   353                        XCHG         ALSO IN DE
005 255   052 055 011                LHLD IPNT    STRING ADDR
005 260   315 163 006                CALL SEAR    SEARCH
005 263   312 303 005                JZ   LMTH    JUMP IF FOUND
005 266   052 067 011                LHLD TEMP    CURRENT ADDR
005 271   043                        INX  H       INCREMENT ADDR
005 272   176                        MOV  A,M     NEXT CHAR
005 273   376 001                    CPI  1       TEST IF EOF
005 275   312 336 005                JZ   BOTM    JUMP IF EOF
005 300   303 250 005                JMP  LNCH    CONTINUE SEARCH
005 303                         *
005 303                         * WHEN FOUND, BACK UP TO START OF LINE
005 303                         *
005 303   052 067 011           LMTH LHLD TEMP    MATCH ADDR
005 306   053                        DCX  H       BACK UP 1
005 307   176                        MOV  A,M     LOAD CHAR
005 310   376 015                    CPI  13      TEST FOR C/R
005 312   302 306 005                JNZ  LMTH+3 LOOP IF NOT C/R
005 315   043                        INX  H       INCREMENT TO LINE START
005 316   042 061 011                SHLD PNTR    SET LINE POINTER
005 321   072 050 011           LMT1 LDA  MODE    OUTPUT MODE FLAG
005 324   267                        ORA  A       SET FLAGS
005 325   302 105 006                JNZ  LIST    PAGE OUTPUT
005 330   315 220 340                CALL CLRS    CLEAR SCREEN
005 333   303 016 003                JMP  LNOT    LINE OUTPUT
005 336                         *
005 336                         * ERROR ROUTINE WHEN EOF REACHED
005 336                         *
005 336   061 200 347           BOTM LXI  SP,STAK       RESET STACK
005 341   052 063 011                LHLD EFPN    EOF ADDRESS
005 344   042 061 011                SHLD PNTR    SET LINE POINTER
005 347   041 360 005                LXI  H,BTMS MESSAGE ADDR
005 352   315 102 002                CALL OUTR    OUTPUT MESSAGE
005 355   303 140 000                JMP  CMRT    RETURN
005 360   102 117 124 124       BTMS DW   'BOTTOM'
005 364   117 115
005 366   215                        DB   8DH
005 367                         *
005 367                         * FIND ROUTINE - COLUMN 1 LOCATE
```

```
005 367                        *
005 367   315 202 005          FIND CALL LOC1   GET STRING LENGTH
005 372   041 074 011               LXI  H,IBUF+2     STRING ADDR
005 375   042 055 011               SHLD IPNT   SET IBUF POINTER
006 000   315 003 002          FIN1 CALL NLST   NEXT LINE ADDR
006 003   042 061 011               SHLD PNTR   SET LINE POINTER
006 006   332 336 005               JC   BOTM   JUMP IF AT EOF
006 011   110                       MOV  C,B    STRING LENGTH TO C
006 012   353                       XCHG        LINE POINTER TO DE
006 013   052 055 011               LHLD IPNT   STRING ADDR
006 016   315 163 006               CALL SEAR   SEARCH
006 021   052 061 011               LHLD PNTR   LINE POINTER
006 024   312 321 005               JZ   LMT1   TO OUTPUT IF MATCH
006 027   303 000 006               JMP  FIN1   CONTINUE SEARCH
006 032                        *
006 032                        * BOTTOM COMMAND ROUTINE
006 032                        *
006 032   052 063 011          BTMM LHLD EFPN   EOF ADDR
006 035   042 061 011               SHLD PNTR   SET LINE POINTER
006 040   311                       RET
006 041                        *
006 041                        * REPLACE COMMAND
006 041                        *
006 041   171                  RLCR MOV  A,C    LINE LENGTH TO A
006 042   326 003                   SUI  3      MINUS 3
006 044   332 020 002               JC   WHAT   ERROR
006 047   015                       DCR  C      CHAR COUNT = INPUT
006 050   015                       DCR  C      MINUS 2
006 051   041 073 011               LXI  H,IBUF+1     STRING ADDR
006 054   042 045 011               SHLD TEM1   SAVE FOR CHANGE ROUTINE
006 057   052 061 011               LHLD PNTR   CURRENT ADDR
006 062   042 067 011               SHLD TEMP   SAVE FOR CHANGE
006 065   006 000                   MVI  B,0    CHARACTER COUNTER
006 067   176                  LNGT MOV  A,M    COUNT CHAR IN CURRENT
006 070   376 015                   CPI  13     LINE UNTIL C/R FOUND
006 072   312 007 005               JZ   CNGO   JUMP TO CHANGE ROUTINE
006 075   332 336 005               JC   BOTM   JUMP IF EOF
006 100   043                       INX  H      INCREMENT ADDR
006 101   004                       INR  B      INCREMENT CHAR COUNT
006 102   303 067 006               JMP  LNGT   LOOP
006 105                        *
006 105                        * PAGE COMMAND
006 105                        *
006 105   052 061 011          LIST LHLD PNTR   CURRENT ADDR
006 110   315 220 340               CALL CLRS   CLEAR SCREEN
006 113   315 076 340               CALL CRLF   C/R OUTPUT
006 116   026 020                   MVI  D,16   LINE COUNTER
006 120   025                  NLS1 DCR  D      DECREMENT LINE COUNT
006 121   310                       RZ          DONE IF ZERO
006 122   315 016 003               CALL LNOT   LINE OUTPUT
006 125   303 120 006               JMP  NLS1   LOOP
006 130                        *
006 130                        * LIST COMMAND
006 130                        *
006 130   315 220 340          LCHR CALL CLRS   CLEAR SCREEN
006 133   052 057 011               LHLD TOPL   TOP ADDRESS
006 136   315 016 003               CALL LNOT   OUTPUT 1 LINE
006 141   322 136 006               JNC  LCHR+6 LOOP IF NOT EOF
006 144   311                       RET
006 145                        *
006 145                        * SCNB ROUTINE - SCAN OFF BLANKS IN IBUF
006 145                        * CARRY SET IF TAB OR C/R FOUND
006 145                        *
006 145   052 055 011          SCNB LHLD IPNT   IBUF POINTER
006 150   176                       MOV  A,M    GET CHARACTER
006 151   376 040                   CPI  ' '    TEST FOR BLANK
```

```
006 153   300                     RNZ          DONE IF NOT BLANK
006 154   043                     INX  H
006 155   042 055 011             SHLD IPNT    UPDATE POINTER
006 160   303 150 006             JMP  SCNB+3 LOOP
006 163                      *
006 163                      * SEAR ROUTINE - STRING SEARCH
006 163                      * AT ADDRESS HL & DE.  LENGTH IN C
006 163                      * ZERO FLAG SET IF MATCH
006 163                      *
006 163   032                SEAR LDAX D       GET CHAR
006 164   276                     CMP  M       TEST FOR MATCH
006 165   300                     RNZ          RETURN IF NO MATCH
006 166   043                     INX  H       INCREMENT ADDRESS
006 167   023                     INX  D
006 170   015                     DCR  C       DECREMENT CHAR COUNT
006 171   302 163 006             JNZ  SEAR    LOOP IF NOT ZERO
006 174   311                     RET
006 175                      *
006 175                      * INAD - INCREMENTS ADDRESS FOR CTSH
006 175                      *
006 175   023                INAD INX  D       INCREMENT ADDRESS
006 176   015                     DCR  C       DECREMENT CHAR COUNT
006 177   302 175 006             JNZ  INAD    LOOP UNTIL ZERO
006 202   014                     INR  C       CLEAR ZERO FLAG
006 203   311                     RET
006 204                      *
006 204                      * CTSH ROUTINE - COMMAND TABLE SEARCH
006 204                      * TABLE ADDRESS IN DE.  NUMBER OF
006 204                      * COMMANDS IN B
006 204                      *
006 204   052 053 011        CTSH LHLD ADDS    LOAD COMMAND ADDR
006 207   072 044 011             LDA  NCHR    COMMAND LENGTH
006 212   117                     MOV  C,A     TO C REGISTER
006 213   315 163 006             CALL SEAR    SEARCH
006 216   304 175 006             CNZ  INAD    INCREMENT IF NO MATCH
006 221   032                     LDAX D       LOW ADDRESS
006 222   157                     MOV  L,A     TO L REG
006 223   023                     INX  D       INCREMENT ADDR
006 224   032                     LDAX D       HIGH ADDRESS
006 225   147                     MOV  H,A     TO H REG
006 226   310                     RZ           RETURN IF MATCH
006 227   023                     INX  D       INCREMENT ADDR
006 230   005                     DCR  B       DECREMENT CMMD COUNT
006 231   302 204 006             JNZ  CTSH    LOOP IF NOT ZERO
006 234   004                     INR  B       CLEAR ZERO FLAG
006 235   311                     RET
006 236                      *
006 236                      * HLIN - ADDRESS INPUT WITH MIN TEST
006 236                      *
006 236   315 103 340        HLIN CALL HL      INPUT ADDRESS
006 241   322 020 002             JNC  WHAT    ERROR IF CARRY CLEAR
006 244   353                     XCHG         ADDR TO DE
006 245   041 202 011             LXI  H,MMIN MIN ADDR
006 250   315 327 003             CALL OVTS    COMPARE
006 253   353                     XCHG         INPUT ADDR TO HL
006 254   320                     RNC          DONE IF MIN OK
006 255   001 273 006             LXI  B,UFMS MESSAGE ADDR
006 260   341                     POP  H       POP WHATEVER'S ON STACK
006 261   041 140 000             LXI  H,CMRT RETURN ADDR
006 264   345                     PUSH H       TO STACK
006 265   041 315 006             LXI  H,UFLL STORAGE AREA
006 270   303 137 002             JMP  EEOF    DISPLAY MIN ADDR
006 273   015                UDMS DB   0FH
006 274   115 111 116 040         DW   'MIN ADDRESS <LH>
006 300   101 104 104 122
006 304   105 123 123 040
```

```
006 310   050 114 110 051
006 314   040
006 315                      UFLL DS   3
006 320   040                     DB   '
006 321                           DS   3
006 324   215                     DB   SOH
006 325                      *
006 325                      * DBCV - CONVERT ASCII NUMBER TO BINARY
006 325                      *
006 325   041 000 000        DBCV LXI  H,0     ZERO PARTIAL RESULT
006 330   012                DBC1 LDAX B       GET NUMBER
006 331   376 015                 CPI  13      TEST FOR C/R
006 333   310                     RZ           DONE IF C/R
006 334   124                     MOV  D,H     PARTIAL RESULT TO DE
006 335   135                     MOV  E,L
006 336   051                     DAD  H       TIMES 2
006 337   051                     DAD  H       TIMES 2
006 340   031                     DAD  D       ADD ORIGINAL PART RES
006 341   051                     DAD  H       TIMES 2; TOTAL=TIMES 10
006 342   326 060                 SUI  48      SUBTRACT ASCII BIAS
006 344   376 012                 CPI  10      TEST FOR VALID #
006 346   077                     CMC
006 347   332 020 002             JC   WHAT    ERROR IF NOT 0 TO 9
006 352   137                     MOV  E,A     NUMBER TO E
006 353   026 000                 MVI  D,0     ZERO D
006 355   031                     DAD  D       ADD TO RESULT
006 356   003                     INX  B       INCREMENT ADDRESS
006 357   303 330 006             JMP  DCB1    LOOP
006 362                      *
006 362                      * TAB ROUTINE - CONVERTS TABS TO SPACES
006 362                      * TAB TO COLUMNS 6, 12, 18, ETC
006 362                      *
006 362   170                TBST MOV  A,B     COLUMN COUNT TO A
006 363   326 006                 SUI  6       SUBTRACT 6 REPEATEDLY
006 365   322 363 006             JNC  $-5     UNTIL OVERFLOW OCCURS
006 370   325                     PUSH D       SAVE D&E
006 371   127                     MOV  D,A     SPACE COUNT(-) TO D
006 372   076 040                 MVI  A,' '   SPACE TO A
006 374   315 052 003             CALL DTOT    OUTPUT SPACE
006 377   004                     INR  B       INCREMENT COLUMN COUNT
007 000   024                     INR  D       INCREMENT SPACE COUNT
007 001   302 374 006             JNZ  $-8     LOOP UNTIL ZERO
007 004   321                     POP  D       RESTORE D&E
007 005   311                     RET
007 006                      *
007 006                      * OUTM COMMAND - OUTPUT DESTINATION SET
007 006                      *
007 006   052 055 011        OUTM LHLD IPNT    IBUF POINTER ADDR
007 011   176                     MOV  A,M     DESTINATION CHAR
007 012   376 123                 CPI  'S'     TEST FOR TVT (SCREEN)
007 014   312 054 007             JZ   TVST    JUMP IF TVT
007 017   376 124                 CPI  'T'     TEST FOR TELETYPE
007 021   302 020 002             JNZ  WHAT    ERROR IF NOT S OR T
007 024   353                     XCHG         SAVE IBUF ADDR
007 025   041 237 010             LXI  H,ABOT TTY ROUTINE ADDR
007 030   042 057 003             SHLD DTO1+1 SET IN OUTPUT ROUTINE
007 033   076 137                 MVI  A,LTCD BAUDOT LETTERS CODE
007 035   062 071 011             STA  MDTY    SET IN TTY MODE FLAG
007 040   315 224 010             CALL SCOT    OUTPUT TO TTY
007 043   353                     XCHG         IBUF ADDR TO HL
007 044   043                     INX  H       INCREMENT ADDR
007 045   176                     MOV  A,M     GET NEXT CHAR
007 046   376 103                 CPI  'C'     TEST IF C
007 050   300                     RNZ          DONE IF NOT C
007 051   303 000 011             JMP  TVCR    C/R L/F OUTPUT TO TTY
007 054   041 071 003        TVST LXI  H,TVTO TVT OUTPUT ROUTINE ADDR
```

```
007 057   042 057 003         SHLD DT01+1 SET IN OUTPUT ROUTINE
007 062   311                 RET
007 063             *
007 063             * BINH - CONVERT BINARY TO OCTAL
007 063             *
007 063   043           BINH INX  H      ENTRY FOR SPACE FIRST
007 064   053                DCX  H      NORMAL ENTRY
007 065   305                PUSH B      SAVE B&C
007 066   016 003            MVI  C,3    LOOP COUNTER
007 070   007                RLC         SHIFT LEFT
007 071   007                RLC         TWO PLACES
007 072   107                MOV  B,A    SAVE VALUE
007 073   241                ANA  C      2 MSB'S IN A NOW
007 074   043           STBN INX  H      INCREMENT STORAGE ADDR
007 075   306 060            ADI  48     ADD ASCII BIAS
007 077   167                MOV  M,A    STORE RESULT
007 100   015                DCR  C      DECREMENT LOOP COUNT
007 101   312 116 007        JZ   BHDN   DONE IF ZERO
007 104   170                MOV  A,B    RECALL VALUE TO A
007 105   007                RLC         SHIFT LEFT
007 106   007                RLC         THREE PLACES
007 107   007                RLC
007 110   107                MOV  B,A    SAVE VALUE
007 111   346 007            ANI  7      MASK THREE BITS
007 113   303 074 007        JMP  STBN   LOOP AND STORE
007 116   301           BHDN POP  B      RESTORE B&C
007 117   311                RET
007 120                 *
007 120                 * PRINT COMMAND
007 120                 *
007 120   016 002       PRTN MVI  C,2    SET LINE COUNT
007 122   332 145 007        JC   PLP1   JUMP IF NO # INPUT
007 125   052 055 011        LHLD IPNT   # ADDR IN IBUF
007 130   104                MOV  B,H    ADDR TO B&C
007 131   115                MOV  C,L
007 132   315 325 006        CALL DBCV   CONVERT TO BINARY
007 135   115                MOV  C,L    RESULT TO C
007 136   045                DCR  H      TEST IF > 255
007 137   362 020 002        JP   WHAT   ERROR IF > 255
007 142   171                MOV  A,C    LINE COUNT TO A
007 143   014                INR  C      INCREMENT COUNT
007 144   267                ORA  A      SET FLAGS
007 145   304 220 340   PLP1 CNZ  CLRS   CLEAR IF NOT ZERO
007 150   052 061 011        LHLD PNTR   CURRENT LINE ADDR
007 153   015           PLP2 DCR  C      DECREMENT LINE COUNT
007 154   310                RZ          DONE IF ZERO
007 155   042 061 011        SHLD PNTR   SET LINE POINTER
007 160   315 016 003        CALL LNOT   OUTPUT 1 LINE
007 163   332 336 005        JC   BOTM   JUMP IF AT EOF
007 166   303 153 007        JMP  PLP2   LOOP
007 171                 *
007 171                 * COMMAND TABLE - COMMAND FOLLOWED
007 171                 * BY ROUTINE ADDRESS
007 171                 *
007 171   105 104 111 124   CTB4 DW   'EDIT'
007 175   354 002                DW   EDCR
007 177   117 125 124 115        DW   'OUTM'
007 203   006 007                DW   OUTM
007 205   103 114 122 123        DW   'CLRS'
007 211   220 340                DW   CLRS
007 213   111 116 123 115        DW   'INSM'
007 217   037 010                DW   IFMC
007 221   120 101 107 105        DW   'PAGE'
007 225   105 006                DW   LIST
007 227   116 105 127 106        DW   'NEWF'
007 233   043 002                DW   NEWF
```

```
007 235   104 111 123 120        DW   'DISP'
007 241   125 002                DW   DCPL
007 243   104 105 117 106        DW   'DEOF'
007 247   203 002                DW   DEOF
007 251   113 111 114 114        DW   'KILL'
007 255   366 001                DW   KLRT
007 257   122 125 102 117        DW   'RUBO'
007 263   303 001                DW   RBCM
007 265   124 101 120 105        DW   'TAPE'
007 271   274 000                DW   TPCR
007 273   102 117 124 115        DW   'BOTM'
007 277   032 006                DW   BTMM
007 301   114 117 101 104        DW   'LOAD'
007 305   303 002                DW   ITCR
007 307   104 111 123 115        DW   'DISM'
007 313   241 002                DW   DISM
007 315   115 117 104 105        DW   'MODE'
007 321   320 001                DW   MSCR
007 323   123 105 124 115        DW   'SETM'
007 327   144 004                DW   MMCR
007 331   114 111 123 124        DW   'LIST'
007 335   130 006                DW   LCHR
007 337                     *
007 337                     * SINGLE CHARACTER COMMANDS
007 337                     *
007 337   101                    DB   'A'
007 340   227 004                DW   APND
007 342   104                    DB   'D'
007 343   334 003                DW   DELE
007 345   111                    DB   'I'
007 346   252 003                DW   INSL
007 350   105                    DB   'E'
007 351   354 002                DW   EDCR
007 353   114                    DB   'L'
007 354   231 005                DW   LOCT
007 356   116                    DB   'N'
007 357   112 001                DW   NXTC
007 361   103                    DB   'C'
007 362   264 004                DW   CCRT
007 364   124                    DB   'T'
007 365   343 001                DW   TOPR
007 367   120                    DB   'P'
007 370   120 007                DW   PRTN
007 372   106                    DB   'F'
007 373   367 005                DW   FIND
007 375   122                    DB   'R'
007 376   041 006                DW   RLCR
010 000   121                    DB   'Q'
010 001   000 340                DW   MONT
010 003                     *
010 003                     * TWO MESSAGES
010 003                     *
010 003   111 116 120 125   INMS DW   'INPUT'
010 007   124
010 010   215                    DB   8DH
010 011   106 125 114 114   EFER DW   'FULL OR PARTIAL FILE?'
010 015   040 117 122 040
010 021   120 101 122 124
010 025   111 101 114 040
010 031   106 111 114 105
010 035   077
010 036   240                    DB   ' '+80H
010 037                     *
010 037.                    * INSM COMMAND - INSERT FROM MEMORY
010 037                     *
010 037   052 055 011       IFMC LHLD IPNT   IBUF ADDR
```

```
010 042   104                    MOV  B,H    TO B&C
010 043   115                    MOV  C,L
010 044   315 325 006            CALL DBCV   # TO BINARY
010 047   045                    DCR  H      TEST VALUE
010 050   362 020 002            JP   WHAT   TOO LARGE
010 053   175                    MOV  A,L
010 054   267                    ORA  A      SET FLAGS
010 055   312 020 002            JZ   WHAT   ERROR, ZERO LINES
010 060   062 044 011            STA  NCHR   SAVE VALUE
010 063   041 075 002            LXI  H,ADMS
010 066   315 102 002            CALL OUTR   OUTPUT MESSAGE
010 071   315 236 006            CALL HLIN   INPUT ADDRESS
010 074   353                    XCHG        TO DE REG
010 075   016 002           IFM4 MVI  C,2    SET CHAR COUNTER
010 077   041 074 011            LXI  H,IBUF+2     SET ADDR
010 102   032               IFM3 LDAX D      GET NEW CHAR
010 103   267                    ORA  A      SET FLAGS
010 104   372 171 010            JM   BADT   BAD IF BIT 8 SET
010 107   376 040                CPI  ' '    TEST IF BLANK
010 111   322 126 010            JNC  IFM1   OK IF >= BLANK
010 114   376 024                CPI  TAB    TEST FOR TAB
010 116   312 126 010            JZ   IFM1   OK IF TAB
010 121   376 015                CPI  13     TEST IF C/R
010 123   302 171 010            JNZ  BADT   BAD IF NOT C/R
010 126                     *
010 126                     * NEW LINE TO IBUF
010 126                     *
010 126   167               IFM1 MOV  M,A    DATA TO IBUF
010 127   043                    INX  H      INCREMENT ADDR
010 130   023                    INX  D
010 131   376 015                CPI  13     TEST FOR C/R
010 133   312 145 010            JZ   IFM2   EOL IF C/R
010 136   014                    INR  C      INCREMENT CHAR COUNT
010 137   171                    MOV  A,C    TO A
010 140   376 110                CPI  CMAX   TEST FOR MAX LENGTH
010 142   302 102 010            JNZ  IFM3   CONTINUE IF OK
010 145   267               IFM2 ORA  A      CLEAR CARRY
010 146   325                    PUSH D      SAVE ADDRESS
010 147   315 252 003            CALL INSL   INSERT LINE
010 152   321                    POP  D
010 153   072 044 011            LDA  NCHR   LOAD LINE COUNTER
010 156   075                    DCR  A      MINUS 1
010 157   062 044 011            STA  NCHR   SAVE LINE COUNT
010 162   302 075 010            JNZ  IFM4   LOOP IF NOT ZERO
010 165   315 076 340            CALL CRLF   C/R TO TVT
010 170   311                    RET
010 171   001 202 010       BADT LXI  B,BDMS MESSAGE ADDR
010 174   041 214 010            LXI  H,BDLC STORAGE ADDR
010 177   303 137 002            JMP  EEOF   OUTPUT BAD DATA ADDR
010 202   015               BDMS DB   0DH
010 203   102 101 104 040        DW   'BAD DATA '
010 207   104 101 124 101
010 213   040
010 214                     BDLC DS   3
010 217   040                    DB   ' '
010 220                          DS   3
010 223   215                    DB   8DH
010 224                     *
010 224                     * OUTPUT TO TTY ROUTINE
010 224                     *
010 224   107               SCOT MOV  B,A    SAVE CHAR
010 225   333 005           SC01 IN   STAT   INPUT UART STATUS
010 227   007                    RLC         TMST FLAG TO CARRY
010 230   322 225 010            JNC  SC01   LOOP UNTIL READY
010 233   170                    MOV  A,B    RECALL DATA
010 234   323 004                OUT  TTY    OUTPUT
```

```
010 236   311                   RET
010 237                    *
010 237                    * ASCII/BAUDOT CONVERSION ROUTINE
010 237                    *
010 237   117              ABOT MOV  C,A    SAVE CHAR
010 240   376 040               CPI  20H    TEST FOR SPACE
010 242   332 341 010           JC   CNTL   CONTROL CHAR IF < SPACE
010 245   302 255 010           JNZ  $+5    JUMP IF NOT SPACE
010 250   076 004               MVI  A,4    BAUDOT SPACE
010 252   303 335 010           JMP  OT1C   OUTPUT
010 255   346 140               ANI  60H    GET BITS 5 & 6
010 257   107                   MOV  B,A    SAVE IN B
010 260   072 071 011           LDA  MDTY   CURRENT TTY MODE
010 263   127                   MOV  D,A    SAVE
010 264   240                   ANA  B      AND WITH CHAR MODE
010 265   302 301 010           JNZ  MDD1   JUMP IF UNCHANGED
010 270   076 144               MVI  A,64H  LOAD 01100100 BINARY
010 272   252                   XRA  D      FORM NEW MODE
010 273   062 071 011           STA  MDTY   SET FLAG
010 276   315 224 010           CALL SCOT   OUTPUT LTRS/FIGS CODE
010 301   076 037          MDD1 MVI  A,31   SET 5 LSB'S
010 303   241                   ANA  C      GET LSB'S OF CHAR
010 304   137                   MOV  E,A    SAVE
010 305   026 000               MVI  D,0
010 307   171                   MOV  A,C    RECALL CHAR
010 310   376 100               CPI  64     TEST IF LTRS OR FIGS
010 312   322 323 010           JNC  LTSH   JUMP IF LTRS
010 315   041 315 341           LXI  H,FIGS FIGS TABLE ADDR
010 320   303 326 010           JMP  $+3
010 323   041 255 341      LTSH LXI  H,LTRS TABLE ADDR
010 326   031                   DAD  D      ADD CHAR ADDRESS
010 327   176                   MOV  A,M    LOAD BAUDOT CHAR
010 330   376 200               CPI  128    TEST IF BIT 8 SET
010 332   322 017 011           JNC  BDEQ   JUMP IF SET
010 335   315 224 010      OT1C CALL SCOT   OUTPUT CHAR
010 340   311                   RET
010 341   376 012          CNTL CPI  10     LINE FEED
010 343   302 353 010           JNZ  $+5
010 346   076 002               MVI  A,BDLF BAUDOT L/F
010 350   303 335 010           JMP  OT1C   OUTPUT
010 353   376 015               CPI  13     C/R
010 355   312 000 011           JZ   TYCR
010 360   376 007               CPI  7      BELL
010 362   300                   RNZ
010 363   076 073               MVI  A,FGCD FIGS CODE
010 365   062 071 011           STA  MDTY   SET FLAG
010 370   315 224 010           CALL SCOT   OUTPUT
010 373   076 005               MVI  A,BDEL BAUDOT BELL
010 375   303 335 010           JMP  OT1C   OUTPUT
011 000   076 010          TYCR MVI  A,BDCR BAUDOT C/R
011 002   315 224 010           CALL SCOT   OUTPUT
011 005   016 006               MVI  C,6    SET C
011 007   315 215 341           CALL TMDL+6 0.3 SEC DELAY
011 012   076 002               MVI  A,BDLF BAUDOT L/F
011 014   303 335 010           JMP  OT1C   OUTPUT
011 017                    *
011 017                    * TWO CHARACTER EQUIVALENCES
011 017                    *
011 017   346 037          BDEQ ANI  1FH    GET 5 LSB'S
011 021   117                   MOV  C,A    SAVE
011 022   006 000               MVI  B,0
011 024   041 355 341           LXI  H,BEQV TABLE ADDR
011 027   011                   DAD  B      ADD OFFSET
011 030   176                   MOV  A,M    GET FIRST CHAR
011 031   345                   PUSH H      SAVE ADDR
011 032   315 237 010           CALL ABOT   OUTPUT 1ST
```

```
011 035  341                   POP  H
011 036  043                   INX  H      NEXT ADDR
011 037  176                   MOV  A,M    SECOND CHAR
011 040  303 237 010           JMP  ABOT   OUTPUT
011 043                   *
011 043                   * VARIABLES
011 043                   *
011 043                   SP   EQU  6      STACK POINTER
011 043                   PSW  EQU  6      STATUS WORD
011 043                   TVT  EQU  0      TVT PORT
011 043                   TAPE EQU  1      TAPE PORT
011 043                   N4CS EQU  17     # OF 4 CHAR CMMDS
011 043                   N1CS EQU  12     # OF 1 CHAR CMMDS
011 043                   CMAX EQU  72     MAX INPUT LENGTH
011 043                   TAB  EQU  20     TAB CHAR
011 043                   CRLF EQU  0E03EH TVT C/R OUTPUT
011 043                   CLRS EQU  0E090H CLEAR SCREEN ROUTINE
011 043                   HL   EQU  0E043H ADDR IN ROUTINE
011 043                   TMDL EQU  0E187H TIME DELAY ROUTINE
011 043                   TAPO EQU  0E169H TAPE OUTPUT ROUTINE
011 043                   TAPI EQU  0E083H TAPE INPUT ROUTINE
011 043                   LTRS EQU  0E1ADH LETTERS TABLE ADDR
011 043                   FIGS EQU  0E1CDH FIGURES TABLE ADDR
011 043                   BEQV EQU  0E1EDH DOUBLE EQUIV TABLE ADDR
011 043                   STAK EQU  0E780H 8080 STACK
011 043                   MONT EQU  0E000H MONITOR ADDR
011 043                   KILL DS   1      KILL CHAR
011 044                   NCHR DS   1      # OF CHAR
011 045                   TEM1 DS   2      TEMPORARY STORAGE
011 047                   RUBO DS   1      RUBOUT FLAG
011 050                   MODE DS   1      OUTPUT MODE FLAG
011 051                   MMAX DS   2      MAX MEMORY POINTER
011 053                   ADDS DS   2      ADDRESS STORAGE
011 055                   IPNT DS   2      IBUF POINTER
011 057                   TOPL DS   2      TOP POINTER
011 061                   PNTR DS   2      LINE POINTER
011 063                   EFPN DS   2      EOF POINTER
011 065                   MVAD DS   2      MOVE ADDR POINTER
011 067                   TEMP DS   2      TEMPORARY STORAGE
011 071                   MDTY DS   1      TTY MODE FLAG
011 072                   IBUF DS   72     INPUT BUFFER
011 202                   BDLF EQU  2      BAUDOT L/F
011 202                   BDCR EQU  8      BAUDOT C/R
011 202                   BDBL EQU  5      BAUDOT BELL
011 202                   FGCD EQU  3BH    FIGURES CODE
011 202                   LTCD EQU  5FH    LETTERS CODE
011 202                   TTY  EQU  4      TTY PORT
011 202                   STAT EQU  5      STATUS PORT
011 202                   MMIN EQU  $      MIN USABLE ADDR
011 202                   *
011 202                   ******* END OF EDITOR
```

APPENDIX F

DISASSEMBLY LISTING OF PILOT EDITOR

The following pages are a symbolic listing of the PILOT Editor. This listing was produced by the Digital Group Z-80 Disassembler, running in the octal, symbolic mode. See C-1 for disassembler operational notes.

```
000000 041 002 340 LD   HL,340002
000003 176         LD   A,(HL)
000004 376 123     CP   123
000006 030 063     JR   063       *000073*
000010 303 004 340 JP   340004
000013 323 002     OUT  002
000015 303 023 000 JP   000023
000020 303 007 340 JP   340007
000023 076 237     LD   A,237
000025 303 033 000 JP   000033
000030 303 012 340 JP   340012
000033 323 001     OUT  001
000035 303 043 000 JP   000043
000040 303 015 340 JP   340015
000043 076 337     LD   A,337
000045 323 001     OUT  001
000047 311         RET
000050 303 020 340 JP   340020
000053 323 001     OUT  001
000055 303 063 000 JP   000063
000060 303 023 340 JP   340023
000063 361         POP  AF
000064 301         POP  BC
000065 321         POP  DE
000066 311         RET
000067 000         NOP
000070 303 026 340 JP   340026
000073 040 005     JR   NZ,005        *000102*
000075 054         INC  L
000076 276         CP   (HL)
000077 312 000 340 JP   Z,340000
000102 061 000 002 LD   SP,002000
000105 021 000 340 LD   DE,340000
000110 076 360     LD   A,360
000112 323 000     OUT  000
000114 315 013 000 CALL 000013
000117 016 010     LD   C,010
000121 333 002     IN   002
000123 241         AND  C
000124 050 373     JR   Z,373        *000121*
000126 076 200     LD   A,200
000130 315 013 000 CALL 000013
000133 315 254 000 CALL 000254
000136 041 000 000 LD   HL,000000
000141 061 000 002 LD   SP,002000
000144 030 003     JR   003       *000151*
000146 303 031 340 JP   340031
000151 076 340     LD   A,340
000153 315 013 000 CALL 000013
000156 315 254 000 CALL 000254
000161 365         PUSH AF
000162 315 254 000 CALL 000254
000165 365         PUSH AF
000166 006 003     LD   B,003
000170 315 254 000 CALL 000254
000173 020 373     DJNZ 373       *000170*
000175 174         LD   A,H
000176 265         OR   L
```

```
000177 040 335      JR   NZ,335       *000136*
000201 361          POP  AF
000202 271          CP   C
000203 372 136 000 JP   N,000136
000206 040 275      JR   NZ,275       *000105*
000210 361          POP  AF
000211 270          CP   B
000212 040 271      JR   NZ,271       *000105*
000214 315 254 000 CALL 000254
000217 022          LD   (DE),A
000220 023          INC  DE
000221 020 371      DJNZ 371       *000214*
000223 315 254 000 CALL 000254
000226 315 254 000 CALL 000254
000231 174          LD   A,H
000232 265          OR   L
000233 040 250      JR   NZ,250       *000105*
000235 333 002      IN   002
000237 346 017      AND  017
000241 040 242      JR   NZ,242       *000105*
000243 014          INC  C
000244 171          LD   A,C
000245 376 020      CP   020
000247 312 001 347 JP   Z,347001
000252 030 275      JR   275       *000151*
000254 325          PUSH DE
000255 305          PUSH BC
000256 026 100      LD   D,100
000260 013          DEC  BC
000261 004          INC  B
000262 020 004      DJNZ 004       *000270*
000264 102          LD   B,D
000265 025          DEC  D
000266 050 215      JR   Z,215       *000105*
000270 333 002      IN   002
000272 346 017      AND  017
000274 050 362      JR   Z,362       *000260*
000276 076 357      LD   A,357
000300 323 001      OUT  001
000302 333 002      IN   002
000304 365          PUSH AF
000305 255          XOR  L
000306 157          LD   L,A
000307 006 007      LD   B,007
000311 027          RLA
000312 255          XOR  L
000313 020 374      DJNZ 374       *000311*
000315 157          LD   L,A
000316 017          RRCA
000317 017          RRCA
000320 137          LD   E,A
000321 346 300      AND  300
000323 254          XOR  H
000324 127          LD   D,A
000325 173          LD   A,E
000326 346 077      AND  077
000330 255          XOR  L
000331 147          LD   H,A
```

```
000332 027          RLA
000333 172          LD   A,D
000334 060 002      JR   NC,002       *000340*
000336 356 001      XOR  001
000340 157          LD   L,A
000341 076 337      LD   A,337
000343 303 053 000 JP   000053
000346 076 177      LD   A,177
000350 315 372 000 CALL 000372
000353 016 004      LD   C,004
000355 006 000      LD   B,000
000357 315 370 000 CALL 000370
000362 020 373      DJNZ 373      *000357*
000364 015          DEC  C
000365 040 370      JR   NZ,370       *000357*
000367 311          RET
000370 076 240      LD   A,240
000372 323 000      OUT  000
000374 257          XOR  A
000375 323 000      OUT  000
000377 311          RET
001000 123          LD   D,E
001001 123          LD   D,E
001002 303 056 023 JP   023056
001005 303 104 011 JP   011104
001010 303 233 011 JP   011233
001013 303 002 006 JP   006002
001016 303 000 011 JP   011000
001021 303 000 003 JP   003000
001024 303 000 000 JP   000000
001027 037          RRA
001030 000          NOP
001031 001 377 011 LD   BC,011377
001034 000          NOP
001035 303 000 000 JP   000000
001040 041 030 001 LD   HL,001030
001043 066 000      LD   (HL),000
001045 054          INC  L
001046 066 001      LD   (HL),001
001050 054          INC  L
001051 066 377      LD   (HL),377
001053 054          INC  L
001054 066 057      LD   (HL),057
001056 041 214 001 LD   HL,001214
001061 315 000 002 CALL 002000
001064 076 145      LD   A,145
001066 323 001      OUT  001
001070 315 173 001 CALL 001173
001073 355          ED   ESCAPE
001074 133 030 001 LD   DE,(001030)
001077 315 124 001 CALL 001124
001102 052 032 001 LD   HL,(001032)
001105 043          INC  HL
001106 023          INC  DE
001107 257          XOR  A
001110 355          ED   ESCAPE
001111 122          SBC  HL,DE
```

```
001112 040 363      JR    NZ,363       *001077*
001114 076 062      LD    A,062
001116 315 173 001  CALL  001173
001121 303 000 000  JP    000000
001124 046 011      LD    H,011
001126 257          XOR   A
001127 032          LD    A,(DE)
001130 027          RLA
001131 323 001      OUT   001
001133 315 155 001  CALL  001155
001136 037          RRA
001137 045          DEC   H
001140 040 367      JR    NZ,367       *001131*
001142 076 001      LD    A,001
001144 323 001      OUT   001
001146 315 155 001  CALL  001155
001151 315 155 001  CALL  001155
001154 311          RET
001155 365          PUSH  AF
001156 072 027 001  LD    A,(001027)
001161 207          ADD   A
001162 207          ADD   A
001163 107          LD    B,A
001164 345          PUSH  HL
001165 000          NOP
001166 020 375      DJNZ  375      *001165*
001170 341          POP   HL
001171 361          POP   AF
001172 311          RET
001173 026 031      LD    D,031
001175 001 003 000  LD    BC,000003
001200 020 376      DJNZ  376      *001200*
001202 015          DEC   C
001203 040 373      JR    NZ,373       *001200*
001205 025          DEC   D
001206 040 365      JR    NZ,365       *001175*
001210 075          DEC   A
001211 040 360      JR    NZ,360       *001173*
001213 311          RET
001214 377          RST   56
001215 154          LD    L,H
001216 327          RST   16
001217 362 351 364  JP    P,364351
001222 351          JP    (HL)
001223 356 347      XOR   347
001225 000          NOP
001226 076 277      LD    A,277
001230 303 124 005  JP    005124
001233 315 346 000  CALL  000346
001236 172          LD    A,D
001237 315 372 005  CALL  005372
001242 315 246 004  CALL  004246
001245 311          RET
001246 260          OR    B
001247 000          NOP
001250 333 000      IN    000
001252 313 177      BIT   7,A
```

```
001254 050 372     JR   Z,372        *001250*
001256 365         PUSH AF
001257 333 000     IN   000
001261 313 177     BIT  7,A
001263 040 372     JR   NZ,372        *001257*
001265 361         POP  AF
001266 311         RET
001267 173         LD   A,E
001270 267         OR   A
001271 036 003     LD   E,003
001273 027         RLA
001274 027         RLA
001275 027         RLA
001276 365         PUSH AF
001277 346 007     AND  007
001301 315 077 002 CALL 002077
001304 361         POP  AF
001305 035         DEC  E
001306 040 363     JR   NZ,363        *001273*
001310 311         RET
001311 000         NOP
001312 000         NOP
001313 000         NOP
001314 104         LD   B,H
001315 002         LD   (BC),A
001316 304 001 050 CALL NZ,050001
001321 173         LD   A,E
001322 344 003 104 CALL PO,104003
001325 002         LD   (BC),A
001326 304 001 010 CALL NZ,010001
001331 214         ADC  H
001332 375         IY   ESCAPE
001333 005         DEC  B
001334 104         LD   B,H
001335 002         LD   (BC),A
001336 304 001 050 CALL NZ,050001
001341 377         RST  56
001342 032         LD   A,(DE)
001343 263         OR   E
001344 032         LD   A,(DE)
001345 265         OR   L
001346 346 240     AND  240
001350 000         NOP
001351 000         NOP
001352 352 003 025 JP   C,000001
001355 240         AND  D
001356 000         NOP
001357 000         NOP
001360 017         RRCA
001361 022         LD   (DE),A
001362 362 014 046 JP   NZ,362014
001365 011         ADD  HL,BC
001366 107         LD   H,(HL)
001367 014         INC  C
001370 046 020     LD   E,014
001372 002         LD   (HL),A
001373 002         EI
001374 253         ADD  E
```

```
001375 001         INC  B
001376 143         INC  HL
001377 025 176     DJNZ 176        *002177*
002001 376 377     CP   377
002003 040 005     JR   NZ,005        *002012*
002005 315 346 000 CALL 000346
002010 030 024     JR   024        *002036*
002012 313 177     BIT  7,A
002014 050 005     JR   Z,005        *002023*
002016 315 372 005 CALL 005372
002021 030 013     JR   013        *002036*
002023 376 000     CP   000
002025 310         RET  Z
002026 365         PUSH AF
002027 315 370 005 CALL 005370
002032 361         POP  AF
002033 075         DEC  A
002034 040 370     JR   NZ,370        *002026*
002036 043         INC  HL
002037 030 337     JR   337        *002000*
002041 315 370 005 CALL 005370
002044 173         LD   A,E
002045 017         RRCA
002046 017         RRCA
002047 017         RRCA
002050 017         RRCA
002051 315 061 002 CALL 002061
002054 173         LD   A,E
002055 315 061 002 CALL 002061
002060 311         RET
002061 346 017     AND  017
002063 376 012     CP   012
002065 070 010     JR   C,010       *002077*
002067 326 011     SUB  011
002071 366 300     OR   300
002073 315 372 005 CALL 005372
002076 311         RET
002077 366 260     OR   260
002101 315 372 005 CALL 005372
002104 311         RET
002105 136         LD   E,(HL)
002106 072 247 001 LD   A,(001247)
002111 376 310     CP   310
002113 312 041 002 JP   Z,002041
002116 303 267 001 JP   001267
002121 377         RST  56
002122 010         EX   AF,AF'
002123 324 326 240 CALL NC,240326
002126 323 324     OUT  324
002130 317         RST  08
002131 322 301 307 JP   NC,307301
002134 305         PUSH BC
002135 240         AND  B
002136 304 325 315 CALL NZ,315325
002141 320         RET  NC
002142 011         ADD  HL,BC
002143 322 345 347 JP   NC,347345
002146 351         JP   (HL)
```

```
002147 363         DI
002150 364 345 362 CALL P,362345
002153 363         DI
002154 272         CP   D
002155 031         ADD  HL,DE
002156 301         POP  BC
002157 003         INC  BC
002160 302 003 303 JP   NZ,303003
002163 003         INC  BC
002164 304 003 305 CALL NZ,305003
002167 003         INC  BC
002170 310         RET  Z
002171 003         INC  BC
002172 314 006 000 CALL Z,000006
002175 006 301     LD   B,301
002177 247         AND  A
002200 002         LD   (BC),A
002201 302 247 002 JP   NZ,002247
002204 303 247 002 JP   002247
002207 304 247 002 CALL NZ,002247
002212 305         PUSH BC
002213 247         AND  A
002214 002         LD   (BC),A
002215 310         RET  Z
002216 247         AND  A
002217 002         LD   (BC),A
002220 314 247 005 CALL Z,005247
002223 000         NOP
002224 043         INC  HL
002225 306 354     ADD  354
002227 341         POP  HL
002230 347         RST  32
002231 363         DI
002232 272         CP   D
002233 034         INC  E
002234 323 240     OUT  240
002236 332 240 310 JP   C,310240
002241 240         AND  B
002242 320         RET  NC
002243 240         AND  B
002244 316 240     ADC  240
002246 303 004 323 JP   323004
002251 247         AND  A
002252 332 247 310 JP   C,310247
002255 247         AND  A
002256 320         RET  NC
002257 247         AND  A
002260 316 247     ADC  247
002262 303 247 005 JP   005247
002265 000         NOP
002266 000         NOP
002267 000         NOP
002270 000         NOP
002271 000         NOP
002272 000         NOP
002273 040 330     JR   NZ,330      *002225*
002275 240         AND  B
002276 311         RET
```

```
002277 356 344     XOR  344
002301 345         PUSH HL
002302 370         RET  M
002303 002         LD   (BC),A
002304 331         EXX
002305 240         AND  B
002306 311         RET
002307 356 344     XOR  344
002311 345         PUSH HL
002312 370         RET  M
002313 003         INC  BC
002314 311         RET
002315 240         AND  B
002316 322 345 347 JP   NC,347345
002321 003         INC  BC
002322 322 240 322 JP   NC,322240
002325 345         PUSH HL
002326 347         RST  32
002327 001 000 041 LD   BC,041000
002332 323 364     OUT  364
002334 341         POP  HL
002335 343         EX   (SP),(HL)
002336 353         EX   DE,(HL)
002337 004         INC  B
002340 322 345 364 JP   NC,364345
002343 365         PUSH AF
002344 362 356 277 JP   P,277356
002347 004         INC  B
002350 311         RET
002351 356 364     XOR  364
002353 345         PUSH HL
002354 362 362 365 JP   P,365362
002357 360         RET  P
002360 364 003 000 CALL P,000003
002363 377         RST  56
002364 305         PUSH BC
002365 356 364     XOR  364
002367 345         PUSH HL
002370 362 240 320 JP   P,320240
002373 341         POP  HL
002374 347         RST  32
002375 345         PUSH HL
002376 240         AND  B
002377 000         NOP
003000 365         PUSH AF
003001 305         PUSH BC
003002 325         PUSH DE
003003 345         PUSH HL
003004 010         EX   AF,AF'
003005 331         EXX
003006 365         PUSH AF
003007 305         PUSH BC
003010 325         PUSH DE
003011 345         PUSH HL
003012 010         EX   AF,AF'
003013 331         EXX
003014 335         IX   ESCAPE
```

```
003015 345         PUSH IX
003016 375         IY   ESCAPE
003017 345         PUSH IY
003020 355         ED   ESCAPE
003021 127         LD   A,I
003022 107         LD   B,A
003023 355         ED   ESCAPE
003024 137         LD   A,R
003025 117         LD   C,A
003026 305         PUSH BC
003027 041 000 000 LD   HL,000000
003032 071         ADD  HL,SP
003033 006 030     LD   B,030
003035 043         INC  HL
003036 020 375     DJNZ 375       *003035*
003040 345         PUSH HL
003041 116         LD   C,(HL)
003042 043         INC  HL
003043 106         LD   B,(HL)
003044 305         PUSH BC
003045 053         DEC  HL
003046 053         DEC  HL
003047 053         DEC  HL
003050 053         DEC  HL
003051 345         PUSH HL
003052 041 121 002 LD   HL,002121
003055 315 000 002 CALL 002000
003060 341         POP  HL
003061 315 240 003 CALL 003240
003064 345         PUSH HL
003065 041 175 002 LD   HL,002175
003070 315 000 002 CALL 002000
003073 341         POP  HL
003074 315 240 003 CALL 003240
003077 353         EX   DE,(HL)
003100 041 224 002 LD   HL,002224
003103 315 000 002 CALL 002000
003106 006 017     LD   B,017
003110 023         INC  DE
003111 020 375     DJNZ 375       *003110*
003113 315 262 003 CALL 003262
003116 006 010     LD   B,010
003120 033         DEC  DE
003121 020 375     DJNZ 375       *003120*
003123 315 262 003 CALL 003262
003126 041 273 002 LD   HL,002273
003131 315 000 002 CALL 002000
003134 006 007     LD   B,007
003136 033         DEC  DE
003137 020 375     DJNZ 375       *003136*
003141 353         EX   DE,(HL)
003142 315 341 003 CALL 003341
003145 315 341 003 CALL 003341
003150 006 003     LD   B,003
003152 315 334 003 CALL 003334
003155 315 341 003 CALL 003341
003160 006 004     LD   B,004
```

```
003162 315 334 003 CALL 003334
003165 006 005     LD   B,005
003167 315 334 003 CALL 003334
003172 345         PUSH HL
003173 041 331 002 LD   HL,002331
003176 315 000 002 CALL 002000
003201 341         POP  HL
003202 315 341 003 CALL 003341
003205 315 341 003 CALL 003341
003210 006 003     LD   B,003
003212 315 334 003 CALL 003334
003215 315 341 003 CALL 003341
003220 006 011     LD   B,011
003222 315 370 005 CALL 005370
003225 020 373     DJNZ 373        *003222*
003227 072 246 001 LD   A,(001246)
003232 315 372 005 CALL 005372
003235 303 346 003 JP   003346
003240 136         LD   E,(HL)
003241 315 106 002 CALL 002106
003244 053         DEC  HL
003245 006 006     LD   B,006
003247 315 370 005 CALL 005370
003252 053         DEC  HL
003253 315 105 002 CALL 002105
003256 020 367     DJNZ 367        *003247*
003260 053         DEC  HL
003261 311         RET
003262 032         LD   A,(DE)
003263 315 316 003 CALL 003316
003266 315 316 003 CALL 003316
003271 315 315 003 CALL 003315
003274 315 315 003 CALL 003315
003277 315 316 003 CALL 003316
003302 315 316 003 CALL 003316
003305 006 003     LD   B,003
003307 315 370 005 CALL 005370
003312 020 373     DJNZ 373        *003307*
003314 311         RET
003315 007         RLCA
003316 007         RLCA
003317 117         LD   C,A
003320 346 001     AND  001
003322 366 260     OR   260
003324 315 372 005 CALL 005372
003327 315 370 005 CALL 005370
003332 171         LD   A,C
003333 311         RET
003334 315 370 005 CALL 005370
003337 020 373     DJNZ 373        *003334*
003341 315 105 002 CALL 002105
003344 053         DEC  HL
003345 311         RET
003346 061 000 002 LD   SP,002000
003351 041 000 000 LD   HL,000000
003354 345         PUSH HL
003355 315 250 001 CALL 001250
```

```
003360 346 337     AND  337
003362 127         LD   D,A
003363 376 200     CP   200
003365 040 003     JR   NZ,003      *003372*
003367 321         POP  DE
003370 030 114     JR   114      *004106*
003372 341         POP  HL
003373 376 322     CP   322
003375 312 000 005 JP   Z,005000
004000 376 310     CP   310
004002 040 006     JR   NZ,006      *004012*
004004 315 233 001 CALL 001233
004007 147         LD   H,A
004010 030 074     JR   074      *004106*
004012 376 314     CP   314
004014 040 006     JR   NZ,006      *004024*
004016 315 233 001 CALL 001233
004021 157         LD   L,A
004022 030 062     JR   062      *004106*
004024 376 214     CP   214
004026 040 003     JR   NZ,003      *004033*
004030 043         INC  HL
004031 030 053     JR   053      *004106*
004033 376 210     CP   210
004035 040 003     JR   NZ,003      *004042*
004037 053         DEC  HL
004040 030 044     JR   044      *004106*
004042 247         AND  A
004043 021 006 000 LD   DE,000006
004046 376 213     CP   213
004050 040 004     JR   NZ,004      *004056*
004052 355         ED   ESCAPE
004053 122         SBC  HL,DE
004054 030 030     JR   030      *004106*
004056 376 212     CP   212
004060 040 004     JR   NZ,004      *004066*
004062 355         ED   ESCAPE
004063 132         ADC  HL,DE
004064 030 020     JR   020      *004106*
004066 366 040     OR   040
004070 365         PUSH AF
004071 006 011     LD   B,011
004073 315 370 005 CALL 005370
004076 020 373     DJNZ 373      *004073*
004100 361         POP  AF
004101 315 251 004 CALL 004251
004104 167         LD   (HL),A
004105 043         INC  HL
004106 345         PUSH HL
004107 315 346 000 CALL 000346
004112 321         POP  DE
004113 325         PUSH DE
004114 142         LD   H,D
004115 173         LD   A,E
004116 376 132     CP   132
004120 060 004     JR   NC,004      *004126*
004122 056 000     LD   L,000
```

```
004124 030 012      JR   012        *034140*
004126 376 264      CP   264
004130 060 004      JR   NC,004        *004136*
004132 056 132      LD   L,132
004134 030 002      JR   002        *004140*
004136 056 264      LD   L,264
004140 134          LD   E,H
004141 315 106 002  CALL 002106
004144 135          LD   E,L
004145 315 106 002  CALL 002106
004150 315 370 005  CALL 005370
004153 315 370 005  CALL 005370
004156 006 006      LD   B,006
004160 321          POP  DE
004161 345          PUSH HL
004162 325          PUSH DE
004163 355          ED   ESCAPE
004164 122          SBC  HL,DE
004165 050 005      JR   Z,005        *004174*
004167 315 370 005  CALL 005370
004172 030 005      JR   005        *004201*
004174 076 232      LD   A,232
004176 315 372 005  CALL 005372
004201 321          POP  DE
004202 341          POP  HL
004203 325          PUSH DE
004204 136          LD   E,(HL)
004205 315 106 002  CALL 002106
004210 043          INC  HL
004211 175          LD   A,L
004212 376 132      CP   132
004214 312 355 003  JP   Z,003355
004217 376 264      CP   264
004221 312 355 003  JP   Z,003355
004224 376 000      CP   000
004226 040 012      JR   NZ,012        *004242*
004230 006 010      LD   B,010
004232 315 370 005  CALL 005370
004235 020 373      DJNZ 373        *004232*
004237 303 355 003  JP   003355
004242 020 314      DJNZ 314        *004160*
004244 030 272      JR   272        *004140*
004246 315 250 001  CALL 001250
004251 107          LD   B,A
004252 072 247 001  LD   A,(001247)
004255 376 310      CP   310
004257 170          LD   A,B
004260 050 044      JR   Z,044        *004326*
004262 315 372 005  CALL 005372
004265 170          LD   A,B
004266 017          RRCA
004267 017          RRCA
004270 346 300      AND  300
004272 117          LD   C,A
004273 315 250 001  CALL 001250
004276 107          LD   B,A
004277 315 372 005  CALL 005372
004302 170          LD   A,B
```

```
004303 007         RLCA
004304 007         RLCA
004305 007         RLCA
004306 346 070     AND  070
004310 201         ADD  C
004311 117         LD   C,A
004312 315 250 001 CALL 001250
004315 107         LD   B,A
004316 315 372 005 CALL 005372
004321 170         LD   A,B
004322 346 007     AND  007
004324 201         ADD  C
004325 311         RET
004326 315 370 005 CALL 005370
004331 170         LD   A,B
004332 315 352 004 CALL 004352
004335 007         RLCA
004336 007         RLCA
004337 007         RLCA
004340 007         RLCA
004341 107         LD   B,A
004342 315 347 004 CALL 004347
004345 200         ADD  B
004346 311         RET
004347 315 250 001 CALL 001250
004352 376 340     CP   340
004354 070 002     JR   C,002          *004360*
004356 326 040     SUB  040
004360 365         PUSH AF
004361 315 372 005 CALL 005372
004364 361         POP  AF
004365 376 272     CP   272
004367 070 002     JR   C,002          *004373*
004371 326 007     SUB  007
004373 326 260     SUB  260
004375 311         RET
004376 000         NOP
004377 000         NOP
005000 061 000 002 LD   SP,002000
005003 355         ED   ESCAPE
005004 106         IM   0
005005 076 260     LD   A,260
005007 062 246 001 LD   (001246),A
005012 373         EI
005013 041 124 005 LD   HL,005124
005016 315 000 002 CALL 002000
005021 315 250 001 CALL 001250
005024 376 272     CP   272
005026 060 371     JR   NC,371          *005021*
005030 376 260     CP   260
005032 070 365     JR   C,365          *005021*
005034 007         RLCA
005035 346 136     AND  136
005037 062 067 005 LD   (005067),A
005042 376 106     CP   106
005044 070 020     JR   C,020          *005066*
005046 376 112     CP   112
005050 060 014     JR   NC,014          *005066*
```

```
005052 376 110     CP   110
005054 060 003     JR   NC,003      *005061*
005056 257         XOR  A
005057 030 002     JR   002      *005063*
005061 076 310     LD   A,310
005063 062 247 001 LD   (001247),A
005066 052 120 005 LD   HL,(005120)
005071 315 346 000 CALL 000346
005074 351         JP   (HL)
005075 000         NOP
005076 000         NOP
005077 000         NOP
005100 300         RET  NZ
005101 005         DEC  B
005102 167         LD   (HL),A
005103 000         NOP
005104 040 001     JR   NZ,001      *005107*
005106 000         NOP
005107 003         INC  BC
005110 000         NOP
005111 003         INC  BC
005112 000         NOP
005113 000         NOP
005114 000         NOP
005115 000         NOP
005116 321         POP  DE
005117 011         ADD  HL,BC
005120 325         PUSH DE
005121 011         ADD  HL,BC
005122 000         NOP
005123 000         NOP
005124 377         RST  56
005125 012         LD   A,(BC)
005126 332 255 270 JP   C,270255
005131 260         OR   B
005132 240         AND  B
005133 317         RST  08
005134 320         RET  NC
005135 240         AND  B
005136 323 331     OUT  331
005140 323 013     OUT  013
005142 317         RST  08
005143 360         RET  P
005144 364 351 357 CALL P,357351
005147 356 363     XOR  363
005151 272         CP   D
005152 070 261     JR   C,261      *005035*
005154 240         AND  B
005155 322 305 301 JP   NC,301305
005160 304 032 262 CALL NZ,262032
005163 240         AND  B
005164 327         RST  16
005165 322 311 324 JP   NC,324311
005170 305         PUSH BC
005171 031         ADD  HL,DE
005172 263         OR   E
005173 240         AND  B
```

```
005174 317         RST  08
005175 343         EX   (SP),(HL)
005176 364 341 354 CALL P,354341
005201 240         AND  B
005202 320         RET  NC
005203 362 357 347 JP   P,347357
005206 362 341 355 JP   P,355341
005211 021 264 240 LD   DE,240264
005214 310         RET  Z
005215 345         PUSH HL
005216 370         RET  N
005217 240         AND  B
005220 320         RET  NC
005221 362 357 347 JP   P,347357
005224 362 341 355 JP   P,355341
005227 063         INC  SP
005230 267         OR   A
005231 240         AND  B
005232 304 351 363 CALL NZ,363351
005235 301         POP  BC
005236 363         DI
005237 363         DI
005240 345         PUSH HL
005241 355         ED   ESCAPE
005242 342         ERR
005243 354 345 362 CALL PE,362345
005246 062 270 240 LD   (240270),A
005251 320         RET  NC
005252 362 351 356 JP   P,356351
005255 364 345 362 CALL P,362345
005260 000         NOP
005261 345         PUSH HL
005262 362 000 000 JP   P,000000
005265 000         NOP
005266 000         NOP
005267 000         NOP
005270 000         NOP
005271 000         NOP
005272 000         NOP
005273 303 324 305 JP   305324
005276 322 000 076 JP   NC,076000
005301 303 062 375 JP   375062
005304 005         DEC  B
005305 257         XOR  A
005306 062 260 005 LD   (005260),A
005311 076 100     LD   A,100
005313 062 133 011 LD   (011133),A
005316 307         RST  00
005317 000         NOP
005320 000         NOP
005321 000         NOP
005322 000         NOP
005323 000         NOP
005324 000         NOP
005325 000         NOP
005326 000         NOP
005327 000         NOP
```

```
005330 072 102 011 LD   A,(011102)
005333 267         OR   A
005334 312 076 011 JP   Z,011076
005337 315 210 011 CALL 011210
005342 315 154 011 CALL 011154
005345 303 076 011 JP   011076
005350 000         NOP
005351 000         NOP
005352 000         NOP
005353 000         NOP
005354 000         NOP
005355 000         NOP
005356 000         NOP
005357 000         NOP
005360 000         NOP
005361 000         NOP
005362 000         NOP
005363 000         NOP
005364 000         NOP
005365 000         NOP
005366 000         NOP
005367 000         NOP
005370 076 240     LD   A,240
005372 315 372 000 CALL 000372
005375 303 370 000 JP   000370
006000 311         RET
006001 000         NOP
006002 000         NOP
006003 303 014 006 JP   006014
006006 303 000 005 JP   005000
006011 000         NOP
006012 000         NOP
006013 000         NOP
006014 345         PUSH HL
006015 325         PUSH DE
006016 305         PUSH BC
006017 365         PUSH AF
006020 346 177     AND  177
006022 376 015     CP   015
006024 050 050     JR   Z,050       *006076*
006026 326 040     SUB  040
006030 070 037     JR   C,037       *006071*
006032 306 300     ADD  300
006034 060 007     JR   NC,007       *006045*
006036 306 345     ADD  345
006040 070 027     JR   C,027       *006071*
006042 326 275     SUB  275
006044 107         LD   B,A
006045 326 270     SUB  270
006047 107         LD   B,A
006050 070 017     JR   C,017       *006071*
006052 072 134 007 LD   A,(007134)
006055 117         LD   C,A
006056 257         XOR  A
006057 052 135 007 LD   HL,(007135)
006062 160         LD   (HL),B
006063 043         INC  HL
```

```
006064 042 135 007 LD   (007135),HL
006067 257         XOR  A
006070 167         LD   (HL),A
006071 361         POP  AF
006072 301         POP  BC
006073 321         POP  DE
006074 341         POP  HL
006075 311         RET
006076 363         DI
006077 315 067 007 CALL 007067
006102 315 053 007 CALL 007053
006105 315 042 007 CALL 007042
006110 076 006     LD   A,006
006112 323 003     OUT  003
006114 333 003     IN   003
006116 313 147     BIT  4,A
006120 040 372     JR   NZ,372        *006114*
006122 315 226 006 CALL 006226
006125 315 122 007 CALL 007122
006130 041 000 005 LD   HL,005000
006133 315 034 007 CALL 007034
006136 315 067 007 CALL 007067
006141 006 140     LD   B,140
006143 041 236 010 LD   HL,010236
006146 076 010     LD   A,010
006150 167         LD   (HL),A
006151 043         INC  HL
006152 020 374     DJNZ 374      *006150*
006154 041 236 010 LD   HL,010236
006157 257         XOR  A
006160 167         LD   (HL),A
006161 042 135 007 LD   (007135),HL
006164 373         EI
006165 315 053 007 CALL 007053
006170 257         XOR  A
006171 323 003     OUT  003
006173 315 053 007 CALL 007053
006176 076 020     LD   A,020
006200 323 003     OUT  003
006202 041 017 020 LD   HL,020017
006205 315 034 007 CALL 007034
006210 315 053 007 CALL 007053
006213 257         XOR  A
006214 323 003     OUT  003
006216 041 141 121 LD   HL,121141
006221 315 034 007 CALL 007034
006224 030 243     JR   243      *006071*
006226 335         IX   ESCAPE
006227 041 236 010 LD   IX,010236
006232 072 134 007 LD   A,(007134)
006235 117         LD   C,A
006236 335         IX   ESCAPE
006237 156 000     LD   L,(IX+000)
006241 076 177     LD   A,177
006243 245         AND  L
006244 157         LD   L,A
006245 257         XOR  A
```

```
006246 147         LD   H,A
006247 265         OR   L
006250 310         RET  Z
006251 006 007     LD   B,007
006253 053         DEC  HL
006254 020 375     DJNZ 375       *006253*
006256 375         IY   ESCAPE
006257 041 134 007 LD   IY,007134
006262 325         PUSH DE
006263 124         LD   D,H
006264 135         LD   E,L
006265 051         ADD  HL,HL
006266 051         ADD  HL,HL
006267 031         ADD  HL,DE
006270 321         POP  DE
006271 353         EX   DE,(HL)
006272 375         IY   ESCAPE
006273 031         ADD  IY,DE
006274 355         ED   ESCAPE
006275 133 137 007 LD   DE,(007137)
006300 325         PUSH DE
006301 353         EX   DE,(HL)
006302 051         ADD  HL,HL
006303 021 357 377 LD   DE,377357
006306 031         ADD  HL,DE
006307 042 132 007 LD   (007132),HL
006312 076 360     LD   A,360
006314 241         AND  C
006315 313 077     SRL  A
006317 313 077     SRL  A
006321 041 036 000 LD   HL,000036
006324 117         LD   C,A
006325 006 000     LD   B,000
006327 011         ADD  HL,BC
006330 321         POP  DE
006331 345         PUSH HL
006332 257         XOR  A
006333 353         EX   DE,(HL)
006334 355         ED   ESCAPE
006335 122         SBC  HL,DE
006336 321         POP  DE
006337 345         PUSH HL
006340 052 132 007 LD   HL,(007132)
006343 315 034 007 CALL 007034
006346 341         POP  HL
006347 016 005     LD   C,005
006351 375         IY   ESCAPE
006352 106 000     LD   B,(IY+000)
006354 072 131 007 LD   A,(007131)
006357 346 200     AND  200
006361 304 000 007 CALL NZ,007000
006364 315 000 007 CALL 007000
006367 375         IY   ESCAPE
006370 043         INC  IY
006371 015         DEC  C
006372 040 355     JR   NZ,355        *006351*
006374 335         IX   ESCAPE
006375 043         INC  IX
```

```
006376 030 232      JR   232        *006232*
007000 325          PUSH DE
007001 345          PUSH HL
007002 076 006      LD   A,006
007004 323 003      OUT  003
007006 076 200      LD   A,200
007010 260          OR   B
007011 323 003      OUT  003
007013 033          DEC  DE
007014 172          LD   A,D
007015 263          OR   E
007016 040 373      JR   NZ,373       *007013*
007020 076 200      LD   A,200
007022 323 003      OUT  003
007024 053          DEC  HL
007025 174          LD   A,H
007026 265          OR   L
007027 040 373      JR   NZ,373       *007024*
007031 341          POP  HL
007032 321          POP  DE
007033 311          RET
007034 053          DEC  HL
007035 175          LD   A,L
007036 264          OR   H
007037 040 373      JR   NZ,373       *007034*
007041 311          RET
007042 365          PUSH AF
007043 333 003      IN   003
007045 313 137      BIT  3,A
007047 050 372      JR   Z,372       *007043*
007051 361          POP  AF
007052 311          RET
007053 365          PUSH AF
007054 076 200      LD   A,200
007056 323 003      OUT  003
007060 046 001      LD   H,001
007062 315 034 007  CALL 007034
007065 361          POP  AF
007066 311          RET
007067 333 003      IN   003
007071 313 147      BIT  4,A
007073 300          RET  NZ
007074 315 053 007  CALL 007053
007077 076 005      LD   A,005
007101 315 042 007  CALL 007042
007104 323 003      OUT  003
007106 333 003      IN   003
007110 313 147      BIT  4,A
007112 050 372      JR   Z,372       *007106*
007114 041 025 006  LD   HL,006025
007117 315 034 007  CALL 007034
007122 315 053 007  CALL 007053
007125 257          XOR  A
007126 323 003      OUT  003
007130 311          RET
007131 000          NOP
007132 357          RST  40
```

```
007133 000             NOP
007134 076 251         LD   A,270
007136 010             EX   AF,AF'
007137 200             ADD  B
007140 000             NOP
007141 200             ADD  B
007142 200             ADD  B
007143 200             ADD  B
007144 200             ADD  B
007145 200             ADD  B
007146 200             ADD  B
007147 200             ADD  B
007150 375             IY   ESCAPE
007151 200             ADD  B
007152 200             ADD  B
007153 200             ADD  B
007154 360             RET  P
007155 200             ADD  B
007156 360             RET  P
007157 200             ADD  B
007160 224             SUB  H
007161 377             RST  56
007162 244             AND  H
007163 377             RST  56
007164 224             SUB  H
007165 222             SUB  D
007166 252             XOR  D
007167 377             RST  56
007170 252             XOR  D
007171 244             AND  H
007172 342 344 210 JP   PO,210344
007175 223             SUB  E
007176 243             AND  E
007177 266             OR   (HL)
007200 311             RET
007201 265             OR   L
007202 202             ADD  D
007203 205             ADD  L
007204 200             ADD  B
007205 200             ADD  B
007206 360             RET  P
007207 200             ADD  B
007210 200             ADD  B
007211 234             SBC  H
007212 242             AND  D
007213 301             POP  BC
007214 200             ADD  B
007215 200             ADD  B
007216 200             ADD  B
007217 200             ADD  B
007220 301             POP  BC
007221 242             AND  D
007222 234             SBC  H
007223 242             AND  D
007224 224             SUB  H
007225 377             RST  56
007226 224             SUB  H
```

```
007227 242         AND  D
007230 210         ADC  B
007231 210         ADC  B
007232 276         CP   (HL)
007233 210         ADC  B
007234 200         ADD  B
007235 200         ADD  B
007236 201         ADD  C
007237 206         ADD  (HL)
007240 200         ADD  B
007241 200         ADD  B
007242 210         ADC  B
007243 210         ADC  B
007244 210         ADC  B
007245 210         ADC  B
007246 210         ADC  B
007247 200         ADD  B
007250 200         ADD  B
007251 201         ADD  C
007252 200         ADD  B
007253 200         ADD  B
007254 202         ADD  D
007255 204         ADD  H
007256 210         ADC  B
007257 220         SUB  B
007260 240         AND  B
007261 276         CP   (HL)
007262 305         PUSH BC
007263 311         RET
007264 321         POP  DE
007265 276         CP   (HL)
007266 200         ADD  B
007267 241         AND  C
007270 377         RST  56
007271 201         ADD  C
007272 201         ADD  C
007273 243         AND  E
007274 305         PUSH BC
007275 311         RET
007276 311         RET
007277 261         OR   C
007300 302 301 311 JP   NZ,311301
007303 331         EXX
007304 346 214     AND  214
007306 224         SUB  H
007307 244         AND  H
007310 377         RST  56
007311 204         ADD  H
007312 362 321 321 JP   P,321321
007315 321         POP  DE
007316 316 236     ADC  236
007320 251         XOR  C
007321 311         RET
007322 311         RET
007323 307         RST  00
007324 300         RET  NZ
```

```
007325 307         RST  00
007326 310         RET  Z
007327 320         RET  NC
007330 340         RET  PO
007331 266         OR   (HL)
007332 311         RET
007333 311         RET
007334 311         RET
007335 266         OR   (HL)
007336 261         OR   C
007337 311         RET
007340 311         RET
007341 312 274 200 JP   Z,200274
007344 200         ADD  B
007345 224         SUB  H
007346 200         ADD  B
007347 200         ADD  B
007350 200         ADD  B
007351 201         ADD  C
007352 226         SUB  (HL)
007353 200         ADD  B
007354 200         ADD  B
007355 210         ADC  B
007356 224         SUB  H
007357 242         AND  D
007360 301         POP  BC
007361 200         ADD  B
007362 224         SUB  H
007363 224         SUB  H
007364 224         SUB  H
007365 224         SUB  H
007366 224         SUB  H
007367 200         ADD  B
007370 301         POP  BC
007371 242         AND  D
007372 224         SUB  H
007373 210         ADC  B
007374 240         AND  B
007375 300         RET  NZ
007376 315 320 240 CALL 240320
010001 276         CP   (HL)
010002 301         POP  BC
010003 335         IX   ESCAPE
010004 315 237 237 CALL 237237
010007 244         AND  H
010010 304 244 237 CALL NZ,237244
010013 377         RST  56
010014 311         RET
010015 311         RET
010016 311         RET
010017 266         OR   (HL)
010020 276         CP   (HL)
010021 301         POP  BC
010022 301         POP  BC
010023 301         POP  BC
010024 343         EX   (SP),(HL)
010025 377         RST  56
```

```
010026 301          POP  BC
010027 301          POP  BC
010030 301          POP  BC
010031 276          CP   (HL)
010032 377          RST  56
010033 311          RET
010034 311          RET
010035 311          RET
010036 301          POP  BC
010037 377          RST  56
010040 310          RET  Z
010041 310          RET  Z
010042 310          RET  Z
010043 300          RET  NZ
010044 276          CP   (HL)
010045 301          POP  BC
010046 301          POP  BC
010047 305          PUSH BC
010050 307          RST  00
010051 377          RST  56
010052 210          ADC  B
010053 210          ADC  B
010054 210          ADC  B
010055 377          RST  56
010056 200          ADD  B
010057 301          POP  BC
010060 377          RST  56
010061 301          POP  BC
010062 200          ADD  B
010063 202          ADD  D
010064 201          ADD  C
010065 201          ADD  C
010066 201          ADD  C
010067 376 377      CP   377
010071 210          ADC  B
010072 224          SUB  H
010073 242          AND  D
010074 301          POP  BC
010075 377          RST  56
010076 201          ADD  C
010077 201          ADD  C
010100 201          ADD  C
010101 201          ADD  C
010102 377          RST  56
010103 240          AND  B
010104 230          SBC  B
010105 240          AND  B
010106 377          RST  56
010107 377          RST  56
010110 220          SUB  B
010111 210          ADC  B
010112 204          ADD  H
010113 377          RST  56
010114 276          CP   (HL)
010115 301          POP  BC
010116 301          POP  BC
010117 301          POP  BC
```

```
010120 276         CP   (HL)
010121 377         RST  56
010122 310         RET  Z
010123 310         RET  Z
010124 310         RET  Z
010125 260         OR   B
010126 276         CP   (HL)
010127 301         POP  BC
010130 305         PUSH BC
010131 302 275 377 JP   NZ,377275
010134 310         RET  Z
010135 314 312 261 CALL Z,261312
010140 262         OR   D
010141 311         RET
010142 311         RET
010143 311         RET
010144 246         AND  (HL)
010145 300         RET  NZ
010146 300         RET  NZ
010147 377         RST  56
010150 300         RET  NZ
010151 300         RET  NZ
010152 376 201     CP   201
010154 201         ADD  C
010155 201         ADD  C
010156 376 374     CP   374
010160 202         ADD  D
010161 201         ADD  C
010162 202         ADD  D
010163 374 377 202 CALL N,202377
010166 214         ADC  H
010167 202         ADD  D
010170 377         RST  56
010171 343         EX   (SP),(HL)
010172 224         SUB  H
010173 210         ADC  B
010174 224         SUB  H
010175 343         EX   (SP),(HL)
010176 340         RET  PO
010177 220         SUB  B
010200 217         ADC  A
010201 220         SUB  B
010202 340         RET  PO
010203 303 305 311 JP   311305
010206 321         POP  DE
010207 341         POP  HL
010210 377         RST  56
010211 377         RST  56
010212 301         POP  BC
010213 301         POP  BC
010214 301         POP  BC
010215 240         AND  B
010216 220         SUB  B
010217 210         ADC  B
010220 204         ADD  H
010221 202         ADD  D
010222 301         POP  BC
```

```
010223 301          POP   BC
010224 301          POP   BC
010225 377          RST   56
010226 377          RST   56
010227 220          SUB   B
010230 220          SUB   B
010231 210          ADC   B
010232 234          SBC   H
010233 210          ADC   B
010234 210          ADC   B
010235 210          ADC   B
010236 030 031      JR    031          *010271*
010240 030 032      JR    032          *010274*
010242 034          INC   E
010243 033          DEC   DE
010244 010          EX    AF,AF'
010245 000          NOP
010246 010          EX    AF,AF'
010247 010          EX    AF,AF'
010250 010          EX    AF,AF'
010251 010          EX    AF,AF'
010252 010          EX    AF,AF'
010253 010          EX    AF,AF'
010254 010          EX    AF,AF'
010255 010          EX    AF,AF'
010256 010          EX    AF,AF'
010257 010          EX    AF,AF'
010260 010          EX    AF,AF'
010261 010          EX    AF,AF'
010262 010          EX    AF,AF'
010263 010          EX    AF,AF'
010264 010          EX    AF,AF'
010265 010          EX    AF,AF'
010266 010          EX    AF,AF'
010267 010          EX    AF,AF'
010270 010          EX    AF,AF'
010271 010          EX    AF,AF'
010272 010          EX    AF,AF'
010273 010          EX    AF,AF'
010274 010          EX    AF,AF'
010275 010          EX    AF,AF'
010276 010          EX    AF,AF'
010277 010          EX    AF,AF'
010300 010          EX    AF,AF'
010301 010          EX    AF,AF'
010302 010          EX    AF,AF'
010303 010          EX    AF,AF'
010304 010          EX    AF,AF'
010305 010          EX    AF,AF'
010306 010          EX    AF,AF'
010307 010          EX    AF,AF'
010310 010          EX    AF,AF'
010311 010          EX    AF,AF'
010312 010          EX    AF,AF'
010313 010          EX    AF,AF'
010314 010          EX    AF,AF'
010315 010          EX    AF,AF'
```

```
010316 010         EX   AF,AF'
010317 010         EX   AF,AF'
010320 010         EX   AF,AF'
010321 010         EX   AF,AF'
010322 010         EX   AF,AF'
010323 010         EX   AF,AF'
010324 010         EX   AF,AF'
010325 010         EX   AF,AF'
010326 010         EX   AF,AF'
010327 010         EX   AF,AF'
010330 010         EX   AF,AF'
010331 010         EX   AF,AF'
010332 010         EX   AF,AF'
010333 010         EX   AF,AF'
010334 010         EX   AF,AF'
010335 010         EX   AF,AF'
010336 010         EX   AF,AF'
010337 010         EX   AF,AF'
010340 010         EX   AF,AF'
010341 010         EX   AF,AF'
010342 010         EX   AF,AF'
010343 010         EX   AF,AF'
010344 010         EX   AF,AF'
010345 010         EX   AF,AF'
010346 010         EX   AF,AF'
010347 010         EX   AF,AF'
010350 010         EX   AF,AF'
010351 010         EX   AF,AF'
010352 010         EX   AF,AF'
010353 010         EX   AF,AF'
010354 010         EX   AF,AF'
010355 010         EX   AF,AF'
010356 010         EX   AF,AF'
010357 010         EX   AF,AF'
010360 010         EX   AF,AF'
010361 010         EX   AF,AF'
010362 010         EX   AF,AF'
010363 010         EX   AF,AF'
010364 010         EX   AF,AF'
010365 010         EX   AF,AF'
010366 010         EX   AF,AF'
010367 010         EX   AF,AF'
010370 010         EX   AF,AF'
010371 010         EX   AF,AF'
010372 010         EX   AF,AF'
010373 010         EX   AF,AF'
010374 010         EX   AF,AF'
010375 010         EX   AF,AF'
010376 000         NOP
010377 000         NOP
011000 365         PUSH AF
011001 305         PUSH BC
011002 325         PUSH DE
011003 345         PUSH HL
011004 366 200     OR   200
011006 376 215     CP   215
011010 312 111 011 JP   Z,011111
```

```
011013 376 212      CP   212
011015 312 076 011 JP   Z,011076
011020 376 000      CP   000
011022 312 076 011 JP   Z,011076
011025 376 233      CP   233
011027 302 052 011 JP   NZ,011052
011032 315 210 011 CALL 011210
011035 052 311 011 LD   HL,(011311)
011040 053          DEC  HL
011041 042 311 011 LD   (011311),HL
011044 315 154 011 CALL 011154
011047 303 076 011 JP   011076
011052 052 311 011 LD   HL,(011311)
011055 365          PUSH AF
011056 072 316 011 LD   A,(011316)
011061 274          CP   H
011062 314 117 011 CALL Z,011117
011065 361          POP  AF
011066 167          LD   (HL),A
011067 043          INC  HL
011070 042 311 011 LD   (011311),HL
011073 315 372 000 CALL 000372
011076 341          POP  HL
011077 321          POP  DE
011100 301          POP  BC
011101 361          POP  AF
011102 347          RST  32
011103 311          RET
011104 076 015      LD   A,015
011106 303 000 011 JP   011000
011111 315 117 011 CALL 011117
011114 303 300 133 JP   133300
011117 315 210 011 CALL 011210
011122 052 313 011 LD   HL,(011313)
011125 353          EX   DE,(HL)
011126 052 315 011 LD   HL,(011315)
011131 114          LD   C,H
011132 041 100 134 LD   HL,134100
011135 176          LD   A,(HL)
011136 022          LD   (DE),A
011137 023          INC  DE
011140 043          INC  HL
011141 174          LD   A,H
011142 271          CP   C
011143 302 135 011 JP   NZ,011135
011146 052 317 011 LD   HL,(011317)
011151 042 311 011 LD   (011311),HL
011154 076 177      LD   A,177
011156 315 372 000 CALL 000372
011161 052 311 011 LD   HL,(011311)
011164 353          EX   DE,(HL)
011165 052 313 011 LD   HL,(011313)
011170 176          LD   A,(HL)
011171 315 372 000 CALL 000372
011174 043          INC  HL
011175 175          LD   A,L
```

```
011176 273         CP   E
011177 302 170 011 JP   NZ,011170
011202 174         LD   A,H
011203 272         CP   D
011204 302 170 011 JP   NZ,011170
011207 311         RET
011210 052 315 011 LD   HL,(011315)
011213 114         LD   C,H
011214 052 311 011 LD   HL,(011311)
011217 174         LD   A,H
011220 271         CP   C
011221 310         RET  Z
011222 066 240     LD   (HL),240
011224 315 370 000 CALL 000370
011227 043         INC  HL
011230 303 217 011 JP   011217
011233 076 337     LD   A,337
011235 315 372 000 CALL 000372
011240 345         PUSH HL
011241 325         PUSH DE
011242 315 210 011 CALL 011210
011245 315 154 011 CALL 011154
011250 321         POP  DE
011251 341         POP  HL
011252 315 250 001 CALL 001250
011255 376 341     CP   341
011257 332 271 011 JP   C,011271
011262 376 373     CP   373
011264 322 271 011 JP   NC,011271
011267 346 337     AND  337
011271 346 177     AND  177
011273 376 177     CP   177
011275 310         RET  Z
011276 376 033     CP   033
011300 312 000 005 JP   Z,005000
011303 376 015     CP   015
011305 310         RET  Z
011306 303 000 011 JP   011000
011311 307         RST  24
011312 137         LD   E,A
011313 000         NOP
011314 134         LD   E,H
011315 000         NOP
011316 140         LD   H,B
011317 300         RET  NZ
011320 137         LD   E,A
011321 257         XOR  A
011322 303 332 011 JP   011332
011325 315 002 006 CALL 006002
011330 076 347     LD   A,347
011332 062 102 011 LD   (011102),A
011335 303 264 016 JP   016264
011340 135         LD   E,L
011341 257         XOR  A
011342 157         LD   L,A
011343 127         LD   D,A
011344 042 311 011 LD   (011311),HL
```

```
011347 042 313 011 LD   (011313),HL
011352 353         EX   DE,(HL)
011353 225         SUB  L
011354 117         LD   C,A
011355 006 377     LD   B,377
011357 051         ADD  HL,HL
011360 051         ADD  HL,HL
011361 051         ADD  HL,HL
011362 051         ADD  HL,HL
011363 031         ADD  HL,DE
011364 042 315 011 LD   (011315),HL
011367 011         ADD  HL,BC
011370 042 317 011 LD   (011317),HL
011373 327         RST  16
011374 000         NOP
011375 000         NOP
011376 000         NOP
011377 000         NOP
012000 303 026 012 JP   012026
012003 041 300 030 LD   HL,030300
012006 315 000 014 CALL 014000
012011 041 200 031 LD   HL,031200
012014 315 000 014 CALL 014000
012017 076 026     LD   A,026
012021 062 001 012 LD   (012001),A
012024 000         NOP
012025 000         NOP
012026 041 030 030 LD   HL,030030
012031 315 000 014 CALL 014000
012034 315 000 015 CALL 015000
012037 303 100 012 JP   012100
012042 000         NOP
012043 000         NOP
012044 000         NOP
012045 000         NOP
012046 041 040 013 LD   HL,013040
012051 315 000 014 CALL 014000
012054 030 350     JR   350       *012026*
012056 000         NOP
012057 000         NOP
012060 000         NOP
012061 000         NOP
012062 000         NOP
012063 000         NOP
012064 000         NOP
012065 000         NOP
012066 000         NOP
012067 000         NOP
012070 000         NOP
012071 000         NOP
012072 000         NOP
012073 104         LD   B,H
012074 015         DEC  C
012075 323 317     OUT  317
012077 326 072     SUB  072
012101 350         RET  PE
012102 133         LD   E,E
```

```
012103 006 012     LD   B,012
012105 041 030 032 LD   HL,032030
012110 276         CP   (HL)
012111 050 010     JR   Z,010      *012123*
012113 043         INC  HL
012114 043         INC  HL
012115 043         INC  HL
012116 020 370     DJNZ 370      *012110*
012120 303 046 012 JP   012046
012123 043         INC  HL
012124 136         LD   E,(HL)
012125 043         INC  HL
012126 126         LD   D,(HL)
012127 353         EX   DE,(HL)
012130 351         JP   (HL)
012131 000         NOP
012132 000         NOP
012133 000         NOP
012134 000         NOP
012135 000         NOP
012136 000         NOP
012137 000         NOP
012140 062 077 012 LD   (012077),A
012143 376 320     CP   320
012145 050 003     JR   Z,003      *012152*
012147 257         XOR  A
012150 030 002     JR   002      *012154*
012152 076 347     LD   A,347
012154 062 102 011 LD   (011102),A
012157 303 026 012 JP   012026
012162 000         NOP
012163 000         NOP
012164 000         NOP
012165 000         NOP
012166 000         NOP
012167 000         NOP
012170 000         NOP
012171 000         NOP
012172 000         NOP
012173 000         NOP
012174 000         NOP
012175 000         NOP
012176 000         NOP
012177 000         NOP
012200 062 076 012 LD   (012076),A
012203 303 026 012 JP   012026
012206 000         NOP
012207 000         NOP
012210 000         NOP
012211 000         NOP
012212 000         NOP
012213 000         NOP
012214 000         NOP
012215 000         NOP
012216 000         NOP
012217 000         NOP
012220 000         NOP
```

```
012221 000         NOP
012222 000         NOP
012223 000         NOP
012224 000         NOP
012225 000         NOP
012226 000         NOP
012227 000         NOP
012230 000         NOP
012231 000         NOP
012232 000         NOP
012233 000         NOP
012234 000         NOP
012235 000         NOP
012236 000         NOP
012237 000         NOP
012240 000         NOP
012241 000         NOP
012242 000         NOP
012243 062 075 012 LD   (012075),A
012246 041 100 015 LD   HL,015100
012251 001 377 100 LD   BC,100377
012254 161         LD   (HL),C
012255 043         INC  HL
012256 020 374     DJNZ 374        *012254*
012260 006 020     LD   B,020
012262 041 100 015 LD   HL,015100
012265 042 073 012 LD   (012073),HL
012270 041 240 013 LD   HL,013240
012273 315 000 016 CALL 016000
012276 376 240     CP   240
012300 050 050     JR   Z,050        *012352*
012302 041 300 013 LD   HL,013300
012305 315 000 016 CALL 016000
012310 052 073 012 LD   HL,(012073)
012313 053         DEC  HL
012314 136         LD   E,(HL)
012315 053         DEC  HL
012316 126         LD   D,(HL)
012317 325         PUSH DE
012320 053         DEC  HL
012321 136         LD   E,(HL)
012322 053         DEC  HL
012323 126         LD   D,(HL)
012324 343         EX   (SP),(HL)
012325 353         EX   DE,(HL)
012326 267         OR   A
012327 355         ED   ESCAPE
012330 122         SBC  HL,DE
012331 070 014     JR   C,014        *012347*
012333 341         POP  HL
012334 042 073 012 LD   (012073),HL
012337 041 340 013 LD   HL,013340
012342 315 000 014 CALL 014000
012345 030 321     JR   321        *012270*
012347 341         POP  HL
012350 020 316     DJNZ 316        *012270*
```

```
012352 303 000 017 JP   017000
012355 315 326 012 CALL 012326
012360 043         INC  HL
012361 106         LD   B,(HL)
012362 315 233 022 CALL 022233
012365 311         RET
012366 006 040     LD   B,040
012370 315 233 022 CALL 022233
012373 311         RET
012374 052 146 040 LD   HL,(040146)
012377 072 006 322 LD   A,(322006)
013002 305         PUSH BC
013003 301         POP  BC
013004 304 331 215 CALL NZ,215331
013007 000         NOP
013010 316 332     ADC  332
013012 332 212 316 JP   C,316212
013015 303 303 212 JP   212303
013020 320         RET  NC
013021 317         RST  08
013022 320         RET  NC
013023 305         PUSH BC
013024 320         RET  NC
013025 212         ADC  D
013026 316 212     ADC  212
013030 000         NOP
013031 000         NOP
013032 000         NOP
013033 000         NOP
013034 000         NOP
013035 000         NOP
013036 000         NOP
013037 000         NOP
013040 035         DEC  E
013041 311         RET
013042 316 326     ADC  326
013044 301         POP  BC
013045 314 311 304 CALL Z,304311
013050 240         AND  B
013051 303 317 315 JP   315317
013054 315 301 316 CALL 316301
013057 304 240 250 CALL NZ,250240
013062 277         CP   A
013063 240         AND  B
013064 306 317     ADD  317
013066 322 240 314 JP   NC,314240
013071 311         RET
013072 323 324     OUT  324
013074 251         XOR  C
013075 215         ADC  L
013076 000         NOP
013077 000         NOP
013100 017         RRCA
013101 311         RET
013102 316 326     ADC  326
013104 301         POP  BC
013105 314 311 304 CALL Z,304311
```

```
013110 240         AND  B
013111 304 305 326 CALL NZ,326305
013114 311         RET
013115 303 305 215 JP   215305
013120 311         RET
013121 254         XOR  H
013122 301         POP  BC
013123 254         XOR  H
013124 311         RET
013125 000         NOP
013126 000         NOP
013127 000         NOP
013130 322 254 301 JP   NC,301254
013133 254         XOR  H
013134 322 000 000 JP   NC,000000
013137 000         NOP
013140 015         DEC  C
013141 311         RET
013142 316 326     ADC  326
013144 301         POP  BC
013145 314 311 304 CALL Z,304311
013150 240         AND  B
013151 315 317 304 CALL 304317
013154 305         PUSH BC
013155 215         ADC  L
013156 000         NOP
013157 000         NOP
013160 000         NOP
013161 000         NOP
013162 000         NOP
013163 000         NOP
013164 000         NOP
013165 000         NOP
013166 000         NOP
013167 000         NOP
013170 000         NOP
013171 000         NOP
013172 000         NOP
013173 000         NOP
013174 000         NOP
013175 000         NOP
013176 000         NOP
013177 000         NOP
013200 022         LD   (DE),A
013201 311         RET
013202 316 326     ADC  326
013204 301         POP  BC
013205 314 311 304 CALL Z,304311
013210 240         AND  B
013211 320         RET  NC
013212 301         POP  BC
013213 322 301 315 JP   NC,315301
013216 305         PUSH BC
013217 324 305 322 CALL NC,322305
013222 215         ADC  L
013223 000         NOP
013224 000         NOP
```

```
013225 000           NOP
013226 000           NOP
013227 000           NOP
013230 000           NOP
013231 000           NOP
013232 000           NOP
013233 000           NOP
013234 000           NOP
013235 000           NOP
013236 000           NOP
013237 000           NOP
013240 007           RLCA
013241 323 324       OUT  324
013243 301           POP  BC
013244 304 304 322   CALL NZ,322304
013247 275           CP   L
013250 000           NOP
013251 000           NOP
013252 000           NOP
013253 000           NOP
013254 000           NOP
013255 000           NOP
013256 000           NOP
013257 000           NOP
013260 000           NOP
013261 000           NOP
013262 000           NOP
013263 000           NOP
013264 000           NOP
013265 000           NOP
013266 000           NOP
013267 000           NOP
013270 000           NOP
013271 000           NOP
013272 000           NOP
013273 000           NOP
013274 000           NOP
013275 000           NOP
013276 000           NOP
013277 000           NOP
013300 007           RLCA
013301 316 304       ADC  304
013303 301           POP  BC
013304 304 304 322   CALL NZ,322304
013307 275           CP   L
013310 000           NOP
013311 000           NOP
013312 000           NOP
013313 000           NOP
013314 000           NOP
013315 000           NOP
013316 000           NOP
013317 000           NOP
013320 000           NOP
013321 000           NOP
013322 000           NOP
013323 000           NOP
```

```
013324 000          NOP
013325 000          NOP
013326 000          NOP
013327 000          NOP
013330 000          NOP
013331 000          NOP
013332 000          NOP
013333 000          NOP
013334 000          NOP
013335 000          NOP
013336 000          NOP
013337 000          NOP
013340 035          DEC  E
013341 311          RET
013342 316 326      ADC  326
013344 301          POP  BC
013345 314 311 304 CALL Z,304311
013350 240          AND  B
013351 301          POP  BC
013352 304 304 322 CALL NZ,322304
013355 305          PUSH BC
013356 323 323      OUT  323
013360 240          AND  B
013361 323 305      OUT  305
013363 324 240 322 CALL NC,322240
013366 305          PUSH BC
013367 255          XOR  L
013370 305          PUSH BC
013371 316 324      ADC  324
013373 305          PUSH BC
013374 322 215 000 JP   NC,000215
013377 000          NOP
014000 305          PUSH BC
014001 106          LD   B,(HL)
014002 043          INC  HL
014003 176          LD   A,(HL)
014004 315 100 014 CALL 014100
014007 043          INC  HL
014010 020 371      DJNZ 371        *014003*
014012 301          POP  BC
014013 311          RET
014014 000          NOP
014015 000          NOP
014016 000          NOP
014017 000          NOP
014020 176          LD   A,(HL)
014021 315 100 014 CALL 014100
014024 043          INC  HL
014025 020 371      DJNZ 371        *014020*
014027 311          RET
014030 365          PUSH AF
014031 076 252      LD   A,252
014033 315 100 014 CALL 014100
014036 361          POP  AF
014037 311          RET
014040 365          PUSH AF
014041 076 240      LD   A,240
```

```
014043 315 100 014 CALL 014100
014046 361         POP  AF
014047 311         RET
014050 365         PUSH AF
014051 076 254     LD   A,254
014053 315 100 014 CALL 014100
014056 361         POP  AF
014057 311         RET
014060 365         PUSH AF
014061 076 250     LD   A,250
014063 315 100 014 CALL 014100
014066 361         POP  AF
014067 311         RET
014070 365         PUSH AF
014071 076 251     LD   A,251
014073 315 100 014 CALL 014100
014076 361         POP  AF
014077 311         RET
014100 376 215     CP   215
014102 040 002     JR   NZ,002        *014106*
014104 327         RST  16
014105 311         RET
014106 357         RST  40
014107 311         RET
014110 315 300 014 CALL 014300
014113 315 227 014 CALL 014227
014116 315 227 014 CALL 014227
014121 136         LD   E,(HL)
014122 345         PUSH HL
014123 026 000     LD   D,000
014125 313 173     BIT  7,E
014127 050 001     JR   Z,001        *014132*
014131 025         DEC  D
014132 043         INC  HL
014133 031         ADD  HL,DE
014134 315 030 014 CALL 014030
014137 315 250 014 CALL 014250
014142 315 030 014 CALL 014030
014145 341         POP  HL
014146 311         RET
014147 000         NOP
014150 000         NOP
014151 000         NOP
014152 000         NOP
014153 000         NOP
014154 000         NOP
014155 000         NOP
014156 000         NOP
014157 000         NOP
014160 365         PUSH AF
014161 076 373     LD   A,373
014163 315 100 014 CALL 014100
014166 361         POP  AF
014167 311         RET
014170 365         PUSH AF
014171 076 375     LD   A,375
014173 315 100 014 CALL 014100
```

```
014176 361         POP  AF
014177 311         RET
014200 315 250 014 CALL 014250
014203 315 040 014 CALL 014040
014206 311         RET
014207 000         NOP
014210 000         NOP
014211 000         NOP
014212 000         NOP
014213 000         NOP
014214 000         NOP
014215 000         NOP
014216 000         NOP
014217 000         NOP
014220 072 076 012 LD   A,(012076)
014223 376 310     CP   310
014225 050 003     JR   Z,003        *014232*
014227 315 040 014 CALL 014040
014232 315 040 014 CALL 014040
014235 315 040 014 CALL 014040
014240 311         RET
014241 000         NOP
014242 000         NOP
014243 000         NOP
014244 000         NOP
014245 000         NOP
014246 000         NOP
014247 000         NOP
014250 325         PUSH DE
014251 345         PUSH HL
014252 355         ED   ESCAPE
014253 133 274 030 LD   DE,(030274)
014256 267         OR   A
014257 355         ED   ESCAPE
014260 122         SBC  HL,DE
014261 174         LD   A,H
014262 315 300 014 CALL 014300
014265 175         LD   A,L
014266 315 300 014 CALL 014300
014271 341         POP  HL
014272 321         POP  DE
014273 311         RET
014274 000         NOP
014275 000         NOP
014276 000         NOP
014277 000         NOP
014300 305         PUSH BC
014301 117         LD   C,A
014302 072 076 012 LD   A,(012076)
014305 376 310     CP   310
014307 050 030     JR   Z,030        *014341*
014311 376 317     CP   317
014313 302 207 012 JP   NZ,012207
014316 006 003     LD   B,003
014320 171         LD   A,C
014321 027         RLA
014322 027         RLA
```

```
014323 027         RLA
014324 365         PUSH AF
014325 346 007     AND  007
014327 306 260     ADD  260
014331 315 100 014 CALL 014100
014334 361         POP  AF
014335 020 362     DJNZ 362       *014321*
014337 301         POP  BC
014340 311         RET
014341 006 002     LD   B,002
014343 171         LD   A,C
014344 007         RLCA
014345 007         RLCA
014346 007         RLCA
014347 007         RLCA
014350 117         LD   C,A
014351 346 017     AND  017
014353 306 260     ADD  260
014355 376 272     CP   272
014357 070 002     JR   C,002       *014363*
014361 306 007     ADD  007
014363 315 100 014 CALL 014100
014366 020 353     DJNZ 353       *014343*
014370 301         POP  BC
014371 311         RET
014372 000         NOP
014373 000         NOP
014374 000         NOP
014375 000         NOP
014376 000         NOP
014377 000         NOP
015000 337         RST  24
015001 376 177     CP   177
015003 050 006     JR   Z,006       *015013*
015005 376 015     CP   015
015007 040 366     JR   NZ,366       *014377*
015011 030 024     JR   024       *015037*
015013 076 033     LD   A,033
015015 357         RST  40
015016 030 360     JR   360       *015000*
015020 076 326     LD   A,326
015022 062 077 012 LD   (012077),A
015025 303 000 012 JP   012000
015030 076 302     LD   A,302
015032 030 366     JR   366       *015022*
015034 000         NOP
015035 000         NOP
015036 000         NOP
015037 000         NOP
015040 000         NOP
015041 305         PUSH BC
015042 325         PUSH DE
015043 345         PUSH HL
015044 006 040     LD   B,040
015046 052 317 011 LD   HL,(011317)
015051 021 340 133 LD   DE,133340
015054 176         LD   A,(HL)
015055 022         LD   (DE),A
```

```
015056 023           INC  DE
015057 043           INC  HL
015060 020 372       DJNZ 372       *015054*
015062 341           POP  HL
015063 321           POP  DE
015064 301           POP  BC
015065 327           RST  16
015066 311           RET
015067 000           NOP
015070 000           NOP
015071 000           NOP
015072 000           NOP
015073 000           NOP
015074 000           NOP
015075 000           NOP
015076 000           NOP
015077 000           NOP
015100 000           NOP
015101 000           NOP
015102 023           INC  DE
015103 377           RST  56
015104 377           RST  56
015105 377           RST  56
015106 377           RST  56
015107 377           RST  56
015110 377           RST  56
015111 377           RST  56
015112 377           RST  56
015113 377           RST  56
015114 377           RST  56
015115 377           RST  56
015116 377           RST  56
015117 377           RST  56
015120 377           RST  56
015121 377           RST  56
015122 377           RST  56
015123 377           RST  56
015124 377           RST  56
015125 377           RST  56
015126 377           RST  56
015127 377           RST  56
015130 377           RST  56
015131 377           RST  56
015132 377           RST  56
015133 377           RST  56
015134 377           RST  56
015135 377           RST  56
015136 377           RST  56
015137 377           RST  56
015140 377           RST  56
015141 377           RST  56
015142 377           RST  56
015143 377           RST  56
015144 377           RST  56
015145 377           RST  56
015146 377           RST  56
015147 377           RST  56
015150 377           RST  56
```

```
015151 377          RST  56
015152 377          RST  56
015153 377          RST  56
015154 377          RST  56
015155 377          RST  56
015156 377          RST  56
015157 377          RST  56
015160 377          RST  56
015161 377          RST  56
015162 377          RST  56
015163 377          RST  56
015164 377          RST  56
015165 377          RST  56
015166 377          RST  56
015167 377          RST  56
015170 377          RST  56
015171 377          RST  56
015172 377          RST  56
015173 377          RST  56
015174 377          RST  56
015175 377          RST  56
015176 377          RST  56
015177 377          RST  56
015200 040 010      JR   NZ,010      *015212*
015202 041 000 030 LD   HL,030000
015205 042 274 015 LD   (015274),HL
015210 000          NOP
015211 000          NOP
015212 052 274 015 LD   HL,(015274)
015215 021 000 030 LD   DE,030000
015220 267          OR   A
015221 355          ED   ESCAPE
015222 122          SBC  HL,DE
015223 042 274 015 LD   (015274),HL
015226 303 026 012 JP   012026
015231 000          NOP
015232 000          NOP
015233 000          NOP
015234 245          AND  L
015235 000          NOP
015236 000          NOP
015237 000          NOP
015240 176          LD   A,(HL)
015241 376 302      CP   302
015243 302 162 030 JP   NZ,030162
015246 303 202 030 JP   030202
015251 000          NOP
015252 000          NOP
015253 000          NOP
015254 000          NOP
015255 000          NOP
015256 000          NOP
015257 000          NOP
015260 000          NOP
015261 000          NOP
015262 000          NOP
015263 000          NOP
```

```
015264 000         NOP
015265 000         NOP
015266 000         NOP
015267 000         NOP
015270 000         NOP
015271 000         NOP
015272 000         NOP
015273 000         NOP
015274 000         NOP
015275 241         AND  C
015276 000         NOP
015277 000         NOP
015300 034         INC  E
015301 323 324     OUT  324
015303 301         POP  BC
015304 322 324 240 JP   NC,240324
015307 303 301 323 JP   323301
015312 323 305     OUT  305
015314 324 324 305 CALL NC,305324
015317 240         AND  B
015320 324 310 305 CALL NC,305310
015323 316 240     ADC  240
015325 250         XOR  B
015326 322 305 324 JP   NC,324305
015331 325         PUSH DE
015332 322 316 251 JP   NC,251316
015335 000         NOP
015336 000         NOP
015337 000         NOP
015340 345         PUSH HL
015341 041 131 022 LD   HL,022131
015344 315 000 014 CALL 014000
015347 327         RST  16
015350 341         POP  HL
015351 315 200 014 CALL 014200
015354 311         RET
015355 000         NOP
015356 000         NOP
015357 000         NOP
015360 000         NOP
015361 000         NOP
015362 000         NOP
015363 000         NOP
015364 000         NOP
015365 000         NOP
015366 000         NOP
015367 000         NOP
015370 000         NOP
015371 000         NOP
015372 000         NOP
015373 000         NOP
015374 000         NOP
015375 000         NOP
015376 000         NOP
015377 000         NOP
016000 042 074 016 LD   (016074),HL
016003 315 000 014 CALL 014000
016006 315 000 015 CALL 015000
```

```
016011 021 347 133 LD   DE,133347
016014 032         LD   A,(DE)
016015 376 240     CP   240
016017 050 021     JR   Z,021       *016042*
016021 016 002     LD   C,002
016023 315 111 016 CALL 016111
016026 070 024     JR   C,024       *016054*
016030 315 100 016 CALL 016100
016033 015         DEC  C
016034 040 365     JR   NZ,365       *016023*
016036 032         LD   A,(DE)
016037 326 240     SUB  240
016041 310         RET  Z
016042 052 073 012 LD   HL,(012073)
016045 313 105     BIT  0,L
016047 040 003     JR   NZ,003       *016054*
016051 313 115     BIT  1,L
016053 310         RET  Z
016054 041 221 016 LD   HL,016221
016057 315 000 014 CALL 014000
016062 041 073 012 LD   HL,012073
016065 313 106     BIT  0,(HL)
016067 052 074 016 LD   HL,(016074)
016072 030 307     JR   307       *016003*
016074 240         AND  B
016075 013         DEC  BC
016076 000         NOP
016077 000         NOP
016100 052 073 012 LD   HL,(012073)
016103 167         LD   (HL),A
016104 043         INC  HL
016105 042 073 012 LD   (012073),HL
016110 311         RET
016111 305         PUSH BC
016112 041 000 000 LD   HL,000000
016115 072 076 012 LD   A,(012076)
016120 376 310     CP   310
016122 050 042     JR   Z,042       *016166*
016124 001 000 003 LD   BC,003000
016127 032         LD   A,(DE)
016130 376 260     CP   260
016132 070 027     JR   C,027       *016163*
016134 376 270     CP   270
016136 060 023     JR   NC,023       *016163*
016140 346 007     AND  007
016142 051         ADD  HL,HL
016143 051         ADD  HL,HL
016144 051         ADD  HL,HL
016145 205         ADD  L
016146 157         LD   L,A
016147 023         INC  DE
016150 020 355     DJNZ 355       *016127*
016152 174         LD   A,H
016153 376 000     CP   000
016155 040 004     JR   NZ,004       *016163*
016157 175         LD   A,L
016160 267         OR   A
```

```
016161 301         POP  BC
016162 311         RET
016163 067         SCF
016164 301         POP  BC
016165 311         RET
016166 001 000 002 LD   BC,002000
016171 032         LD   A,(DE)
016172 326 260     SUB  260
016174 376 012     CP   012
016176 070 006     JR   C,006       *016206*
016200 326 007     SUB  007
016202 376 020     CP   020
016204 060 355     JR   NC,355       *016163*
016206 051         ADD  HL,HL
016207 051         ADD  HL,HL
016210 051         ADD  HL,HL
016211 051         ADD  HL,HL
016212 205         ADD  L
016213 157         LD   L,A
016214 023         INC  DE
016215 020 352     DJNZ 352       *016171*
016217 030 336     JR   336       *016157*
016221 027         RLA
016222 301         POP  BC
016223 304 304 322 CALL NZ,322304
016226 305         PUSH BC
016227 323 323     OUT  323
016231 240         AND  B
016232 305         PUSH BC
016233 322 322 317 JP   NC,317322
016236 322 240 322 JP   NC,322240
016241 305         PUSH BC
016242 255         XOR  L
016243 305         PUSH BC
016244 316 324     ADC  324
016246 305         PUSH BC
016247 322 215 043 JP   NC,043215
016252 327         RST  16
016253 315 023 013 CALL 013023
016256 311         RET
016257 357         RST  40
016260 030 367     JR   367       *016251*
016262 000         NOP
016263 000         NOP
016264 303 327 016 JP   016327
016267 041 000 033 LD   HL,033000
016272 106         LD   B,(HL)
016273 257         XOR  A
016274 167         LD   (HL),A
016275 206         ADD  (HL)
016276 302 306 016 JP   NZ,016306
016301 160         LD   (HL),B
016302 043         INC  HL
016303 303 272 016 JP   016272
016306 045         DEC  H
016307 045         DEC  H
016310 045         DEC  H
016311 045         DEC  H
016312 072 133 011 LD   A,(011133)
016315 157         LD   L,A
```

```
016316 042 133 011 LD   (011133),HL
016321 041 327 016 LD   HL,016327
016324 042 265 016 LD   (016265),HL
016327 052 133 011 LD   HL,(011133)
016332 303 340 011 JP   011340
016335 000         NOP
016336 000         NOP
016337 000         NOP
016340 000         NOP
016341 315 223 013 CALL 013223
016344 315 005 016 CALL 016005
016347 042 116 040 LD   (040116),HL
016352 041 141 040 LD   HL,040141
016355 176         LD   A,(HL)
016356 267         OR   A
016357 302 365 016 JP   NZ,016365
016362 041 135 040 LD   HL,040135
016365 042 120 040 LD   (040120),HL
016370 353         EX   DE,(HL)
016371 041 014 040 LD   HL,040014
016374 315 126 016 CALL 016126
016377 052 041 100 LD   HL,(100041)
017002 015         DEC  C
017003 042 376 017 LD   (017376),HL
017006 052 376 017 LD   HL,(017376)
017011 355         ED   ESCAPE
017012 133 073 012 LD   DE,(012073)
017015 267         OR   A
017016 355         ED   ESCAPE
017017 122         SBC  HL,DE
017020 322 026 012 JP   NC,012026
017023 052 376 017 LD   HL,(017376)
017026 126         LD   D,(HL)
017027 043         INC  HL
017030 136         LD   E,(HL)
017031 043         INC  HL
017032 315 000 030 CALL 030000
017035 000         NOP
017036 325         PUSH DE
017037 126         LD   D,(HL)
017040 043         INC  HL
017041 136         LD   E,(HL)
017042 043         INC  HL
017043 042 376 017 LD   (017376),HL
017046 315 013 030 CALL 030013
017051 000         NOP
017052 341         POP  HL
017053 072 075 012 LD   A,(012075)
017056 376 304     CP   304
017060 312 300 017 JP   Z,017300
017063 376 323     CP   323
017065 312 000 020 JP   Z,020000
017070 315 200 014 CALL 014200
017073 006 010     LD   B,010
017075 176         LD   A,(HL)
017076 117         LD   C,A
017077 346 177     AND  177
017101 376 177     CP   177
017103 050 077     JR   Z,077         *017204*
017105 376 015     CP   015
017107 050 121     JR   Z,121         *017232*
```

```
017111 376 012      CP   012
017113 050 122      JR   Z,122        *017237*
017115 376 033      CP   033
017117 050 123      JR   Z,123        *017244*
017121 376 040      CP   040
017123 050 124      JR   Z,124        *017251*
017125 000          NOP
017126 000          NOP
017127 000          NOP
017130 000          NOP
017131 313 171      BIT  7,C
017133 040 040      JR   NZ,040        *017175*
017135 315 040 014  CALL 014040
017140 171          LD   A,C
017141 366 200      OR   200
017143 315 100 014  CALL 014100
017146 315 040 014  CALL 014040
017151 043          INC  HL
017152 020 321      DJNZ 321        *017075*
017154 327          RST  16
017155 042 374 017  LD   (017374),HL
017160 355          ED   ESCAPE
017161 133 372 017  LD   DE,(017372)
017164 353          EX   DE,(HL)
017165 267          OR   A
017166 355          ED   ESCAPE
017167 122          SBC  HL,DE
017170 070 214      JR   C,214        *017006*
017172 353          EX   DE,(HL)
017173 030 273      JR   273        *017070*
017175 076 255      LD   A,255
017177 315 100 014  CALL 014100
017202 030 334      JR   334        *017140*
017204 021 314 304  LD   DE,304314
017207 313 171      BIT  7,C
017211 040 010      JR   NZ,010        *017223*
017213 173          LD   A,E
017214 306 040      ADD  040
017216 137          LD   E,A
017217 172          LD   A,D
017220 306 040      ADD  040
017222 127          LD   D,A
017223 113          LD   C,E
017224 172          LD   A,D
017225 315 100 014  CALL 014100
017230 030 306      JR   306        *017140*
017232 021 322 303  LD   DE,303322
017235 030 350      JR   350        *017207*
017237 021 306 314  LD   DE,314306
017242 030 343      JR   343        *017207*
017244 021 301 302  LD   DE,302301
017247 030 336      JR   336        *017207*
017251 021 320 323  LD   DE,323320
017254 030 331      JR   331        *017207*
017256 000          NOP
017257 000          NOP
017260 000          NOP
```

```
017261 000          NOP
017262 000          NOP
017263 000          NOP
017264 000          NOP
017265 000          NOP
017266 000          NOP
017267 000          NOP
017270 000          NOP
017271 000          NOP
017272 000          NOP
017273 000          NOP
017274 000          NOP
017275 000          NOP
017276 000          NOP
017277 000          NOP
017300 315 200 014  CALL 014200
017303 006 010      LD   B,010
017305 072 077 012  LD   A,(012077)
017310 376 326      CP   326
017312 040 012      JR   NZ,012       *017326*
017314 072 076 012  LD   A,(012076)
017317 376 310      CP   310
017321 050 003      JR   Z,003       *017326*
017323 327          RST  16
017324 030 003      JR   003       *017331*
017326 315 040 014  CALL 014040
017331 176          LD   A,(HL)
017332 315 300 014  CALL 014300
017335 043          INC  HL
017336 020 366      DJNZ 366       *017326*
017340 327          RST  16
017341 042 374 017  LD   (017374),HL
017344 353          EX   DE,(HL)
017345 052 372 017  LD   HL,(017372)
017350 267          OR   A
017351 355          ED   ESCAPE
017352 122          SBC  HL,DE
017353 332 006 017  JP   C,017006
017356 353          EX   DE,(HL)
017357 030 317      JR   317       *017300*
017361 000          NOP
017362 000          NOP
017363 000          NOP
017364 000          NOP
017365 000          NOP
017366 000          NOP
017367 000          NOP
017370 000          NOP
017371 000          NOP
017372 377          RST  56
017373 023          INC  DE
017374 374 017 104  CALL N,104017
017377 015          DEC  C
020000 315 200 014  CALL 014200
020003 176          LD   A,(HL)
020004 376 100      CP   100
020006 332 300 022  JP   C,022300
```

```
020011 376 200     CP   200
020013 332 220 020 JP   C,020220
020016 376 300     CP   300
020020 332 200 020 JP   C,020200
020023 303 300 020 JP   020300
020026 000         NOP
020027 000         NOP
020030 000         NOP
020031 000         NOP
020032 000         NOP
020033 000         NOP
020034 000         NOP
020035 000         NOP
020036 000         NOP
020037 000         NOP
020040 043         INC  HL
020041 042 374 017 LD   (017374),HL
020044 041 370 017 LD   HL,017370
020047 176         LD   A,(HL)
020050 313 107     BIT  0,A
020052 040 004     JR   NZ,004        *020060*
020054 257         XOR  A
020055 167         LD   (HL),A
020056 030 003     JR   003        *020063*
020060 313 207     RES  0,A
020062 167         LD   (HL),A
020063 327         RST  16
020064 052 374 017 LD   HL,(017374)
020067 353         EX   DE,(HL)
020070 052 372 017 LD   HL,(017372)
020073 267         OR   A
020074 355         ED   ESCAPE
020075 122         SBC  HL,DE
020076 332 006 017 JP   C,017006
020101 353         EX   DE,(HL)
020102 030 274     JR   274        *020000*
020104 000         NOP
020105 000         NOP
020106 000         NOP
020107 000         NOP
020110 345         PUSH HL
020111 076 003     LD   A,003
020113 220         SUB  B
020114 365         PUSH AF
020115 176         LD   A,(HL)
020116 315 300 014 CALL 014300
020121 315 040 014 CALL 014040
020124 043         INC  HL
020125 020 366     DJNZ 366        *020115*
020127 361         POP  AF
020130 050 011     JR   Z,011        *020143*
020132 107         LD   B,A
020133 315 220 014 CALL 014220
020136 315 040 014 CALL 014040
020141 020 370     DJNZ 370        *020133*
020143 305         PUSH BC
020144 341         POP  HL
```

```
020145 051         ADD  HL,HL
020146 051         ADD  HL,HL
020147 001 000 023 LD   BC,023000
020152 011         ADD  HL,BC
020153 006 004     LD   B,004
020155 176         LD   A,(HL)
020156 315 100 014 CALL 014100
020161 043         INC  HL
020162 020 371     DJNZ 371       *020155*
020164 315 040 014 CALL 014040
020167 341         POP  HL
020170 311         RET
020171 000         NOP
020172 000         NOP
020173 000         NOP
020174 000         NOP
020175 000         NOP
020176 000         NOP
020177 000         NOP
020200 006 001     LD   B,001
020202 315 240 024 CALL 024240
020205 176         LD   A,(HL)
020206 346 007     AND  007
020210 315 260 024 CALL 024260
020213 303 040 020 JP   020040
020216 000         NOP
020217 000         NOP
020220 376 166     CP   166
020222 040 011     JR   NZ,011       *020235*
020224 001 027 001 LD   BC,001027
020227 315 110 020 CALL 020110
020232 303 040 020 JP   020040
020235 001 020 001 LD   BC,001020
020240 072 370 017 LD   A,(017370)
020243 346 006     AND  006
020245 050 001     JR   Z,001       *020250*
020247 004         INC  B
020250 315 110 020 CALL 020110
020253 176         LD   A,(HL)
020254 365         PUSH AF
020255 346 070     AND  070
020257 037         RRA
020260 037         RRA
020261 037         RRA
020262 315 260 024 CALL 024260
020265 315 050 014 CALL 014050
020270 361         POP  AF
020271 346 007     AND  007
020273 315 260 024 CALL 024260
020276 030 332     JR   332       *020232*
020300 313 107     BIT  0,A
020302 312 000 021 JP   Z,021000
020305 313 117     BIT  1,A
020307 312 100 021 JP   Z,021100
020312 313 127     BIT  2,A
020314 312 367 021 JP   Z,021367
020317 001 010 001 LD   BC,001010
```

```
020322 315 110 020 CALL 020110
020325 176         LD   A,(HL)
020326 346 070     AND  070
020330 050 013     JR   Z,013      *020345*
020332 037         RRA
020333 037         RRA
020334 037         RRA
020335 107         LD   B,A
020336 016 010     LD   C,010
020340 257         XOR  A
020341 201         ADD  C
020342 047         DAA
020343 020 374     DJNZ 374      *020341*
020345 117         LD   C,A
020346 006 002     LD   B,002
020350 171         LD   A,C
020351 007         RLCA
020352 007         RLCA
020353 007         RLCA
020354 007         RLCA
020355 117         LD   C,A
020356 346 017     AND  017
020360 306 260     ADD  260
020362 315 100 014 CALL 014100
020365 020 361     DJNZ 361      *020350*
020367 303 040 020 JP   020040
020372 000         NOP
020373 000         NOP
020374 000         NOP
020375 000         NOP
020376 000         NOP
020377 000         NOP
021000 313 117     BIT  1,A
021002 050 045     JR   Z,045      *021051*
021004 313 127     BIT  2,A
021006 050 015     JR   Z,015      *021025*
021010 006 002     LD   B,002
021012 315 240 024 CALL 024240
021015 043         INC  HL
021016 176         LD   A,(HL)
021017 315 300 014 CALL 014300
021022 303 040 020 JP   020040
021025 001 017 003 LD   BC,003017
021030 315 110 020 CALL 020110
021033 176         LD   A,(HL)
021034 315 170 024 CALL 024170
021037 315 050 014 CALL 014050
021042 043         INC  HL
021043 315 100 024 CALL 024100
021046 303 040 020 JP   020040
021051 313 127     BIT  2,A
021053 050 005     JR   Z,005      *021062*
021055 001 021 003 LD   BC,003021
021060 030 346     JR   346      *021030*
021062 001 015 001 LD   BC,001015
021065 315 110 020 CALL 020110
021070 176         LD   A,(HL)
021071 315 170 024 CALL 024170
```

```
021074 303 040 020 JP   020040
021077 000         NOP
021100 313 137     BIT  3,A
021102 040 054     JR   NZ,054       *021160*
021104 313 127     BIT  2,A
021106 050 034     JR   Z,034       *021144*
021110 001 011 001 LD   BC,001011
021113 315 110 020 CALL 020110
021116 176         LD   A,(HL)
021117 346 070     AND  070
021121 037         RRA
021122 037         RRA
021123 037         RRA
021124 117         LD   C,A
021125 072 370 017 LD   A,(017370)
021130 346 007     AND  007
021132 062 370 017 LD   (017370),A
021135 171         LD   A,C
021136 315 000 025 CALL 025000
021141 303 040 020 JP   020040
021144 001 012 001 LD   BC,001012
021147 030 342     JR   342       *021113*
021151 000         NOP
021152 000         NOP
021153 000         NOP
021154 000         NOP
021155 000         NOP
021156 000         NOP
021157 000         NOP
021160 376 311     CP   311
021162 050 076     JR   Z,076       *021262*
021164 376 331     CP   331
021166 050 103     JR   Z,103       *021273*
021170 376 351     CP   351
021172 050 104     JR   Z,104       *021300*
021174 376 371     CP   371
021176 050 126     JR   Z,126       *021326*
021200 376 315     CP   315
021202 050 146     JR   Z,146       *021352*
021204 376 335     CP   335
021206 050 045     JR   Z,045       *021255*
021210 376 355     CP   355
021212 312 036 027 JP   Z,027036
021215 001 013 005 LD   BC,005013
021220 072 370 017 LD   A,(017370)
021223 313 107     BIT  0,A
021225 302 040 020 JP   NZ,020040
021230 260         OR   B
021231 000         NOP
021232 062 370 017 LD   (017370),A
021235 006 001     LD   B,001
021237 315 110 020 CALL 020110
021242 345         PUSH HL
021243 041 131 022 LD   HL,022131
021246 315 000 014 CALL 014000
021251 341         POP  HL
```

```
021252 303 040 020 JP   020040
021255 001 014 003 LD   BC,003014
021260 030 336     JR   336      *021220*
021262 001 015 001 LD   BC,001015
021265 315 110 020 CALL 020110
021270 303 040 020 JP   020040
021273 001 016 001 LD   BC,001016
021276 030 365     JR   365      *021265*
021300 001 017 001 LD   BC,001017
021303 315 110 020 CALL 020110
021306 072 370 017 LD   A,(017370)
021311 366 020     OR   020
021313 062 370 017 LD   (017370),A
021316 076 004     LD   A,004
021320 315 100 032 CALL 032100
021323 303 040 020 JP   020040
021326 001 020 001 LD   BC,001020
021331 315 110 020 CALL 020110
021334 021 370 017 LD   DE,017370
021337 032         LD   A,(DE)
021340 366 040     OR   040
021342 022         LD   (DE),A
021343 076 006     LD   A,006
021345 315 120 032 CALL 032120
021350 030 351     JR   351      *021323*
021352 001 021 003 LD   BC,003021
021355 315 110 020 CALL 020110
021360 043         INC  HL
021361 315 100 024 CALL 024100
021364 303 040 020 JP   020040
021367 376 303     CP   303
021371 040 005     JR   NZ,005      *022000*
021373 001 017 003 LD   BC,003017
021376 030 355     JR   355      *021355*
022000 376 313     CP   313
022002 312 160 022 JP   Z,022160
022005 376 323     CP   323
022007 040 011     JR   NZ,011      *022022*
022011 001 022 002 LD   BC,002022
022014 315 110 020 CALL 020110
022017 303 015 021 JP   021015
022022 376 333     CP   333
022024 040 005     JR   NZ,005      *022033*
022026 001 023 002 LD   BC,002023
022031 030 361     JR   361      *022014*
022033 376 343     CP   343
022035 040 020     JR   NZ,020      *022057*
022037 001 024 001 LD   BC,001024
022042 315 110 020 CALL 020110
022045 345         PUSH HL
022046 041 117 022 LD   HL,022117
022051 315 000 014 CALL 014000
022054 341         POP  HL
022055 030 236     JR   236      *021315*
022057 376 353     CP   353
022061 040 014     JR   NZ,014      *022077*
022063 001 024 001 LD   BC,001024
```

```
022066 315 110 020 CALL 020110
022071 345         PUSH HL
022072 041 125 022 LD   HL,022125
022075 030 352     JR   352         *022051*
022077 376 363     CP   363
022101 040 006     JR   NZ,006         *022111*
022103 001 025 001 LD   BC,001025
022106 303 265 021 JP   021265
022111 001 026 001 LD   BC,001026
022114 303 265 021 JP   021265
022117 005         DEC  B
022120 250         XOR  B
022121 323 320     OUT  320
022123 251         XOR  C
022124 254         XOR  H
022125 003         INC  BC
022126 304 305 254 CALL NZ,254305
022131 006 305     LD   B,305
022133 323 303     OUT  303
022135 301         POP  BC
022136 320         RET  NC
022137 305         PUSH BC
022140 043         INC  HL
022141 176         LD   A,(HL)
022142 315 300 014 CALL 014300
022145 303 040 017 JP   017040
022150 000         NOP
022151 000         NOP
022152 000         NOP
022153 000         NOP
022154 000         NOP
022155 000         NOP
022156 000         NOP
022157 000         NOP
022160 345         PUSH HL
022161 335         IX   ESCAPE
022162 341         POP  IX
022163 006 002     LD   B,002
022165 072 370 017 LD   A,(017370)
022170 346 006     AND  006
022172 050 003     JR   Z,003         *022177*
022174 004         INC  B
022175 335         IX   ESCAPE
022176 043         INC  IX
022177 335         IX   ESCAPE
022200 176 001     LD   A,(IX+001)
022202 376 100     CP   100
022204 070 051     JR   C,051         *022257*
022206 346 300     AND  300
022210 007         RLCA
022211 007         RLCA
022212 075         DEC  A
022213 306 042     ADD  042
022215 117         LD   C,A
022216 315 110 020 CALL 020110
022221 335         IX   ESCAPE
022222 176 001     LD   A,(IX+001)
022224 346 070     AND  070
```

```
022226 017         RRCA
022227 017         RRCA
022230 017         RRCA
022231 306 260     ADD  260
022233 315 100 014 CALL 014100
022236 315 050 014 CALL 014050
022241 335         IX   ESCAPE
022242 176 001     LD   A,(IX+001)
022244 346 007     AND  007
022246 315 260 024 CALL 024260
022251 043         INC  HL
022252 000         NOP
022253 000         NOP
022254 303 040 020 JP   020040
022257 335         IX   ESCAPE
022260 176 001     LD   A,(IX+001)
022262 346 070     AND  070
022264 017         RRCA
022265 017         RRCA
022266 017         RRCA
022267 306 045     ADD  045
022271 117         LD   C,A
022272 315 110 020 CALL 020110
022275 030 342     JR   342        *022241*
022277 000         NOP
022300 346 007     AND  007
022302 312 200 025 JP   Z,025200
022305 376 001     CP   001
022307 312 320 025 JP   Z,025320
022312 376 002     CP   002
022314 312 040 026 JP   Z,026040
022317 376 006     CP   006
022321 332 167 025 JP   C,025167
022324 312 260 026 JP   Z,026260
022327 176         LD   A,(HL)
022330 346 070     AND  070
022332 017         RRCA
022333 017         RRCA
022334 017         RRCA
022335 306 030     ADD  030
022337 117         LD   C,A
022340 006 001     LD   B,001
022342 315 110 020 CALL 020110
022345 303 040 020 JP   020040
022350 000         NOP
022351 000         NOP
022352 000         NOP
022353 000         NOP
022354 000         NOP
022355 000         NOP
022356 000         NOP
022357 000         NOP
022360 001 055 001 LD   BC,001055
022363 315 110 020 CALL 020110
022366 176         LD   A,(HL)
022367 315 300 014 CALL 014300
022372 303 040 020 JP   020040
```

```
022375 000         NOP
022376 000         NOP
022377 000         NOP
023000 301         POP  BC
023001 304 304 240 CALL NZ,240304
023004 301         POP  BC
023005 304 303 240 CALL NZ,240303
023010 323 325     OUT  325
023012 302 240 323 JP   NZ,323240
023015 302 303 240 JP   NZ,240303
023020 301         POP  BC
023021 316 304     ADC  304
023023 240         AND  B
023024 330         RET  C
023025 317         RST  08
023026 322 240 317 JP   NC,317240
023031 322 240 240 JP   NC,240240
023034 303 320 240 JP   240320
023037 240         AND  B
023040 322 323 324 JP   NC,324323
023043 240         AND  B
023044 320         RET  NC
023045 325         PUSH DE
023046 323 310     OUT  310
023050 320         RET  NC
023051 317         RST  08
023052 320         RET  NC
023053 240         AND  B
023054 311         RET
023055 331         EXX
023056 240         AND  B
023057 240         AND  B
023060 311         RET
023061 330         RET  C
023062 240         AND  B
023063 240         AND  B
023064 322 305 324 JP   NC,324305
023067 240         AND  B
023070 305         PUSH BC
023071 330         RET  C
023072 330         RET  C
023073 240         AND  B
023074 312 320 240 JP   Z,240320
023077 240         AND  B
023100 314 304 240 CALL Z,240304
023103 240         AND  B
023104 303 301 314 JP   314301
023107 314 317 325 CALL Z,325317
023112 324 240 311 CALL NC,311240
023115 316 240     ADC  240
023117 240         AND  B
023120 305         PUSH BC
023121 330         RET  C
023122 240         AND  B
023123 240         AND  B
023124 304 311 240 CALL NZ,240311
023127 240         AND  B
023130 305         PUSH BC
```

```
023131 311         RET
023132 240         AND  B
023133 240         AND  B
023134 310         RET  Z
023135 301         POP  BC
023136 314 324 322 CALL Z,322324
023141 314 303 301 CALL Z,301303
023144 322 322 303 JP   NC,303322
023147 301         POP  BC
023150 322 314 301 JP   NC,301314
023153 240         AND  B
023154 322 322 301 JP   NC,301322
023157 240         AND  B
023160 304 301 301 CALL NZ,301301
023163 240         AND  B
023164 303 320 314 JP   314320
023167 240         AND  B
023170 323 303     OUT  303
023172 306 240     ADD  240
023174 303 303 306 JP   306303
023177 240         AND  B
023200 311         RET
023201 316 303     ADC  303
023203 240         AND  B
023204 304 305 303 CALL NZ,303305
023207 240         AND  B
023210 302 311 324 JP   NZ,324311
023213 240         AND  B
023214 322 305 323 JP   NC,323305
023217 240         AND  B
023220 323 305     OUT  305
023222 324 240 322 CALL NC,322240
023225 314 303 240 CALL Z,240303
023230 322 322 303 JP   NC,303322
023233 240         AND  B
023234 322 314 240 JP   NC,240314
023237 240         AND  B
023240 322 322 240 JP   NC,240322
023243 240         AND  B
023244 323 314     OUT  314
023246 301         POP  BC
023247 240         AND  B
023250 323 322     OUT  322
023252 301         POP  BC
023253 240         AND  B
023254 305         PUSH BC
023255 322 322 240 JP   NC,240322
023260 323 322     OUT  322
023262 314 240 325 CALL Z,325240
023265 316 313     ADC  313
023267 316 312     ADC  312
023271 322 240 240 JP   NC,240240
023274 316 317     ADC  317
023276 320         RET  NC
023277 240         AND  B
023300 304 312 316 CALL NZ,316312
023303 332 305 304 JP   C,304305
```

```
023306 240         AND  B
023307 240         AND  B
023310 314 304 311 CALL Z,311304
023313 240         AND  B
023314 303 320 311 JP   311320
023317 240         AND  B
023320 311         RET
023321 316 311     ADC  311
023323 240         AND  B
023324 317         RST  08
023325 325         PUSH DE
023326 324 311 314 CALL NC,314311
023331 304 304 240 CALL NZ,240304
023334 303 320 304 JP   304320
023337 240         AND  B
023340 311         RET
023341 316 304     ADC  304
023343 240         AND  B
023344 317         RST  08
023345 325         PUSH DE
023346 324 304 314 CALL NC,314304
023351 304 311 322 CALL NZ,322311
023354 303 320 311 JP   311320
023357 322 311 316 JP   NC,316311
023362 311         RET
023363 322 317 324 JP   NC,324317
023366 311         RET
023367 322 314 304 JP   NC,304314
023372 304 322 303 CALL NZ,303322
023375 320         RET  NC
023376 304 322 311 CALL NZ,311322
```

APPENDIX G

PILOT AUTHOR SYSTEM USER MANUAL

TABLE OF CONTENTS

1.0 INTRODUCTION

This PILOT facility is designed to support authors in the creation of PILOT lessons. It includes the capabilities necessary to prepare a PILOT lesson, save that lesson on a digital cassette tape, perform a trial execution of that lesson, modify the lesson if required, and finally prepare a cassette tape of the lesson for student use. The PILOT language is extremely simple to use, and authors with no prior computer experience typically find they can prepare lessons with little difficulty.

2.0 AUTHOR SYSTEM

Before continuing, the term AUTHOR SYSTEM must be defined. This term simply means the collection of hardware, software and procedures that an author will need to perform the function of authoring PILOT lessons. The principle elements of this system are covered in the following paragraphs.

2.1 Hardware Operations

The hardware for this PILOT system consists of a Digital Group Inc. Z-80 Microprocessor computer system. Input/output devices required are an output display, a keyboard, and a digital cassette mass storage system with one or more drives. The PILOT AUTHOR SYSTEM is also designed to support a Digital Group matrix printer but this device is not absolutely required for operation.

The author needs to know very little about the operation of this system. His main concern is setting up the proper tape for the author function and turning the system on. An additional operation that will be required is the disabling of the computer's read-only-memory. This will be discussed in detail in proper sequence of operation.

The author should be familiar with four hardware items. The first of these is the ON/OFF switch. This is the red button located on the front panel of the computer in the lower right hand corner. Push it in to turn the computer on. The button will light up when it is pushed. If it stays lit, the computer is on. Push it again to turn the computer off. See the special note below about turning the computer off.

The next hardware item of concern is the cassette tape drive. This device operates under computer control; therefore, there are no controls for STOP, PLAY, REWIND and FAST-FORWARD that are on most cassette players. With the exception of the lack of controls, it is very similar to familiar cassette tape recorders and should pose no problems. The tape is loaded by placing the rear of the cassette in first and pushing down firmly on the front of the cassette. There is a small black lever on the right, front edge of the cassette drive that will pop the tape out for removal. CAUTION: One of the few problems that can occur in correctly operating hardware is a tape tangle. With few exceptions, this is caused by turning off the computer power while the tape is reading forward. Prior to turning off the computer, check to be sure the tape is not moving.

The other hardware items of concern are the location of the Read Only Memory (ROM) disable switch and the printer ON/OFF switch. Request the computer support personnel to show you the location of these switches as they vary from system to system. The use of these switches will be covered later.

2.2 PHIMON Operations

PHIMON stands for Phi Deck Monitor. Phi Deck is the manufacturer's name for the computer control digital cassette tape drive. PHIMON serves as the operating system for the Digital Group hardware. In simple terms, it is the software that operates the hardware. An author will give PHIMON a command and PHIMON will manipulate the hardware to execute that command.

For example, if an author wishes to know what are the lessons stored in a particular tape, he inserts the tape in the cassette and gives PHIMON the command "DIRECTORY." PHIMON will then rewind the tape to the beginning and then read the tape. Once PHIMON has read in the names of the lessons on the tape, it will stop the tape and display the names of the lessons. As PHIMON is working, it places abbreviated messages on the display. These are notifications as to what it is doing at a particular moment. RD OVLY means that it is <u>reading</u> in some of its own software and <u>overlaying</u> this new software in memory on top of software it does not need right now. RD DIR means it is <u>reading</u> a <u>directory</u>. These messages can be ignored for now. Later they will be a comfort as they inform the author that the system is working properly.

The functions that PHIMON will perform for authors are loading of files from tape into memory, starting the execution of programs, saving of data in memory out onto tape as a file, and moving files from one tape to another.

Each of these functions will be discussed in the sequence of use in the author´s role of creating lessons.

2.3 PILOT Interpreter Operations

The PILOT interpreter is the software element that performs the actions the author has placed in the lesson instructions. If the authors have placed an instruction in the PILOT lesson to have the computer display textual material to the student, the PILOT interpreter reads the lesson finding the author´s command, then displays the text as requested. The author can have the computer display text, ask the student a question and analyze the student´s answer by matching on keywords or phrases. If the author desires, he can have the computer load a new lesson from the tape.

A key point of this relationship is that the author must insure that the computer has instructions in the form of a PILOT lesson. If the PILOT interpreter is started without a lesson loaded, the computer will run without control and will have to be stopped. It is very simple to insure this does not happen and techniques for dealing with this situation will be presented in later sections of this manual.

2.4 PILOT Editor Operations

The PILOT editor is the software component that authors use to create PILOT lessons. The general sequence used is to run the editor, select the mode of operation (Display Only or Display and Hardcopy), instruct the editor where to place the lesson in memory, enter the lesson and finally save the lesson on tape.

While entering the lesson, the author has the capability to make changes in a variety of ways. Appendix G.A, PILOT Editor Commands, of this users manual gives a detailed discussion of all editor commands and file operations.

3.0 PILOT LANGUAGE

3.1 Statement Types

The principle feature of PILOT is its simplicity. The instruction format is one or more letters followed by a colon to define the statement type.

Example:

```
T:  HELLO!  Welcome to PILOT.
```

This sequence instructs PILOT to display Text. The instructions T: for display Text, A: for accept Answer and M: for Match answer are used most often. These can be put together for a simple program.

```
T:    HAVE YOU USED PILOT BEFORE?

A:

M:    YES, YEA, YEP, SURE, RIGHT

TY:   THEN YOU DON'T HAVE TO GO THRU
      THIS INTRODUCTION

TN:   THEN YOU WILL NEED THIS INTRODUCTION
```

The letters Y and N are called conditioners. They are added to a code to make it conditional on the previous match. The following is an example of how the above program will execute (the output from the computer in all examples will be in upper case; lower case will be used to indicate user input through the keyboard).

```
                HAVE YOU USED PILOT BEFORE?

     yes i have

                 THEN YOU DON'T HAVE TO GO THRU

                       THIS INTRODUCTION
```

If the user input had not matched any of the words after M:, then the computer would display:

THEN YOU WILL NEED THIS INTRODUCTION.

There are five more statements types commonly used in PILOT lessons. J: stands for jump. It means jump to a new place in the program. U: stands for use. It means to use a segment of the lesson as a subroutine. E: stands for end. It marks the end of a subroutine or the end of the entire lesson. C: means compute. Only assignment, addition and subtraction are used. Legal values for variables are integers in the range from +99 to -99. Variables are specified by single letters A through Z. R: means remark. This statement allows authors to put remarks into the lessons for documentation.

The conditioners of Y and N can be used by themselves as an abbreviated form of TN: or TY:. Numeric variables can also be used as conditioners if placed in parentheses after the instruction but before the colon. If this conditioner's value is greater than zero, the statement will be execution. If both Y:, N: and numeric variables are used for conditioners, the numeric variable will override the Y: N: conditioner. The M: statement can be modified by adding a C after it and before the colon. This modification permits the search of text with commas. Details of matching operations will be discussed in later paragraphs.

Continuation of T:, Y:, or N: statements can be made by starting the continued line by a colon.

J: and U: statements must be used in conjunction with labels. Labels are formed in PILOT by placing an asterisk before 1 to 10 characters. Labels maybe duplicated. PILOT will find the first of the duplicated labels if they are referenced.

PILOT supports numeric and string variables. Numeric variables begin with the pound sign (#) followed by of a single letter. If a numeric variable is used as a conditioner or in C: statement, the # character is not used (i.e., C: A=A+1). Numeric variables may range from +99 to -99 integer values. They may be used in A: statements to allow the student to enter a numeric answer. Numeric variables can also be used in T: statement. In this latter case, the current value of the variable is printed. Numeric variable values may be altered by A: statements or C: statements assigning a new value. They are all set to zero at the start up of PILOT.

String variables can be used in A: statements or T: statements. The name of a string variable must begin with a dollar sign ($) and can have one to ten characters. Blanks are not allowed in string variable names.

For example:

```
                                 (Student Taking
       (Lesson)                      Lesson)

    T: WHAT'S YOUR NAME?         WHAT'S YOUR NAME
    A: $NAME                     larry
    T: HELLO, $NAME.             HELLO, LARRY.
```

If there is no value assigned to the string variable, the name of the string variable is displayed.

There are four more commands that are in this version of PILOT. The first of these is "NEW$:". This command erases the values in all string variables. The next command is "INMAX:". INMAX: specifies the maximun number of characters that the A: statement will accept.

For example:

```
            (Lesson)

        T: ENTER A TWO DIGIT NUMBER
        INMAX:  2
        A:
```

In the above example, the student cannot enter more than two digits. When PILOT starts, INMAX: is automatically set to a value of 72.

The next command is the LOAD: command. This command is used to load a new lesson into the PILOT interpreter and start execution of the new lesson. The load command has two formats. To load a new lesson from the system tape cassette (tape drive zero), the format is:

```
LOAD:(blank)(lesson name)
```

For example, to load the lesson "INTRO" from drive zero:

```
LOAD: INTRO
```

Note that there must be a blank between the colon and the lesson name.

If the lesson to be loaded is on a nonsystem drive (drive 1, 2 or 3), the format is as follows:

```
LOAD:(#)(drive number)(blank)(lesson name)
```

For example, to load the lesson TEXT1 from drive one:

```
LOAD:#1 TEXT1
```

Note that there must be a blank between the drive number and the lesson name.

Error conditions are reported to the students with brief messages. If the student makes a non-numeric response to an A: statement with a numeric variable, the PILOT interpreter will display "NUMERIC ENTRY REQUIRED." If an attempt is made to place a value in a string variable that already has a value assigned, the error message "NO ROOM" will result. An error message "NO ROOM" is typically the result of an error in lesson logic rather than student error.

If a nonexistent label is referenced, the error message "_LABEL NOT FOUND" will result. This happens when using the U: (USE) or J: (JUMP) statements. If a numeric expression will result in a value greater than +99 or less than -99, the expression will be displayed and the error message "*ILLEGAL EXPRESSION" followed by either "*VALUE > 99" or "*VALUE < -99".

The loading of a new lesson can cause the error message "OVERFLOW" if there is insufficient room for the new lesson. Since lessons and string variables share the same memory, sometimes the size of lesson space can be increased by using the NEW$: statement to clear all string variables before loading the new lesson.

The last command to be needed in PILOT is the "BYE:" statement. When the interpreter encounters the "BYE:"

statement, control of the computer is turned over to the PHIMON operating system.

The matching operation in PILOT is simple. The PILOT interpreter scans the last student input of an A: statement for a match with any of the patterns in the M: statement. If any items match, the YN condition is set to Y. Patterns in the M: statement are separated by commas. Single blanks can be part of the pattern. A comma can be used after the last pattern to indicate trailing blanks. If there is no comma after the last pattern, the pattern ends with the last nonblank character.

The MC: statement is used to perform matching on student inputs containing commas. It requires that the caret (^) be used in place of the comma as the pattern separator.

Student inputs are scanned prior to the matching operation to reduce multiple blanks to one blank and to add a blank to beginning and end of the response. Multiple blanks are also reduced to one blank in all patterns before matching.

EXAMPLES

```
     M:X,Y,Z
             Matches    ZEROX,    ZOO,  or   BOY
                       (Z) (X)   (Z)        (Y)


     M: X, Y, Z
             Matches    ZEROX    or    ZOO
                       (Z)            (Z)
```

```
          Does not match BOY


      M: X , Y , Z ,
          Matches only X or Y or Z
```

3.2 Sample Programs

The purpose of this section is to present short program segments of PILOT lessons to illustrate the PILOT language. These samples do not necessarily represent efficient coding, rather they are to present each of the PILOT language statement types. The T: (Text), A: (Answer) and M: (Match) statements are the basic building blocks of PILOT.

```
SAMPLE-------------------------------------------------------------

           (Lesson)                      (Student Execution)

 T: HELLO.  WHATS YOUR NAME?        HELLO.  WHATS YOUR NAME?
 A: $NAME                           larry
 T: WELCOME TO PILOT $NAME          WELCOME TO PILOT LARRY
  : HAVE YOU USED PILOT BEFORE      HAVE YOU USED PILOT BEFORE?
 A:                                 yes i have
 M: YES, YEP, YEA,
TY: GOOD, LETS GET ON               GOOD, LETS GET ON
  : WITH YOUR LESSON                WITH YOUR LESSON
TN: WOULD YOU LIKE
  : AN INTRODUCTION
  : TO PILOT?

-------------------------------------------------------------------
```

The use of T: A: and M: statements in this sample program is facilitated by the string varible "$NAME". The colons after the T: statements are continuation lines of the preceding statement.

The next samples will illustrate the U: (use as subroutine) and J: (Jump) statements.

```
SAMPLE--------------------------------------------------

T: WELCOME TO PILOT
 : WOULD YOU LIKE TO REVIEW
 : HOW TO USE PILOT?
U: *YES
Y: OK.  LETS START OUR REVIEW
 : FROM THE BEGINNING.
J: *REVIEW
N: OK.  I GUESS YOU ARE
 : AN OLD TIMER AT THIS
J: *MAIN_LESSON
*  YES
A:
M: YES, YEP, YEA, OK, SURE, GUESS, RIGHT
E:

--------------------------------------------------------
```

The subroutine *YES can be used to analyze the answer to any question that has an affirmative response. "Sure why not," "I guess so" and "all right" would be recognized by this subroutine. E: marks the end of the subroutine. Once the answer is analyzed, the J: statement jumps to either *REVIEW (not in this sample) or *MAIN_LESSON (not in this sample).

The C: (compute) statement allows keeping score on answers, number of times through a loop, branching and many other operations. The following sample illustrates each of these operations.

```
SAMPLE--------------------------------------------------

   R: "W" IS THE NUMBER OF WRONG ANSWERS.
   C: W=W+1
   T: THIS IS THE #W TIME
```

```
    : YOU HAVE HAD AN OPPORTUNITY
    : TO ANSWER THIS QUESTION NAME$.
   R: "L" IS THE LIMIT WRONG ANSWERS
   C: Z=L-W
   R: IF Z > 0 THEN JUMP TO * AGAIN
J(Z): * AGAIN
   T: I AM SORRY NAME$, BUT
    : NO MORE TIME ON THAT
    : QUESTION.  LETS GO ON
    : TO SOMETHING DIFFERENT.

--------------------------------------------------------------
```

The R: (remarks) statement is used to comment on the function of the numeric variables in this example and explain how the numeric variable works as a conditioner.

The LOAD command is used to load and begin execution of a new PILOT lesson. The important thing to remember is that the new lesson must be on the proper tape drive or the system will not be able to find the lesson and errors will result.

```
SAMPLE-------------------------------------------------------

   T: ENTER DESIRED LESSON NAME
    * START
   A:
   M: PASCAL
  JY: *PASCAL
   M: PILOT
  JY: *PILOT
   M: COBOL
  JY: *COBOL
   M: SPARKS
  JY: *SPARKS
   M: BASIC
  JY: *BASIC
   T: INCORRECT LESSON NAME.
    : I CAN TEACH:
    :   PASCAL
    :   PILOT
    :   COBOL
    :   SPARKS
    :   BASIC
    : ---------
    : PLEASE SELECT AGAIN
   C: T=T+1
   C: L=2
   C: Z=L-T
J(L) * WRONG_LESSON
   R: T=TIMES THRU MENUE
   R: IF SECOND TIME, MUST BE IN
   R: WRONG LESSON
   J: *START
    * PASCAL
   U: *WAIT_MSG
    LOAD:#1 PASCAL
    * COBOL
   U: *WAIT_MSG
    LOAD:#1 COBOL
    * PILOT
   U: *WAIT_MSG
    LOAD: PILOT
    * SPARKS
   U: *WAIT_MSG
    LOAD#1 SPARKS
    * BASIC
   U: *WAIT_MSG
    LOAD:#1 BASIC

    * WAIT_MSG
   T: PLEASE WAIT ONE MINUTE
    : WHILE I LOAD YOUR
    : LESSON.
   E:
```

The load command cannot be made conditional by placing a conditioner with it. The above sample illustrates a PILOT lesson on the system tape (TAPE ZERO) and all other lessons on tape 1. Note that R: statements cannot be continued with colons.

4.0 SAMPLE AUTHOR SESSION

This section will present a complete author session on the computer. The functions covered will begin with turning on the hardware through final preparation of a tape for student use.

Start by placing a tape in drive zero and pressing the red ON/OFF button (lower right corner). The system should respond as follows:

1. Red light of ON/OFF switch lights and stays lighted.
2. Tape Drive Zero rewinds at high speed to the beginning, head engages and tape moves forward slowly.
3. After a few seconds the display should output message:

   "RD OVLY"

4. Display should output message:

   "RD DIR"

   and tape in Drive Zero should rewind to beginning, then start forward again.
5. Tape should stop with head disengaged. Display should output directory of all files on the system tape.

If step two does not work correctly, press the ON/OFF switch again insuring that it goes all the way in before it is released. If the red light still does not come on, call for help.

If step one does not work correctly, check to insure the "ROM Disable Switch" is in the off position. If it was in the on position, turn the power off by pushing in red ON/OFF button, place the ROM Disable Switch in the off position and start over from the beginning.

If the files directory comes on the display properly, check the list of file names to insure "PEDIT" and "PILOT" files are on the tape. If they are not, call for help.

4.1 Create Lesson

Place the "ROM Disable Switch" in the on position. Give the command:

"RUN PEDIT"

This will bring into execution the PILOT EDITOR. Once the program PEDIT has been loaded into memory and is running, the following message will be displayed:

```
------------------------------------------------------------

                         Z-80 PILOT
                           EDITOR

1.  EDIT PILOT LESSON - DISPLAY

2.  EDIT PILOT LESSON - HARD COPY

3.  RETURN TO PHIMON

          MAKE SELECTION PLEASE

------------------------------------------------------------
```

Selection of choice one (by entering the number 1) will start the edit process with the keyboard as input and the display as output. Selection of choice two will place the output on the printer and on the display. If choice two is desired, insure the printer is turned on before selecting choice two or the system will not respond. If this does happen, press the reset button (blue button next to the red ON/OFF switch). This will return the system to PHIMON and the PEDIT program can be restarted by entering the command "START."

Assuming choice one is selected, the display will output

```
------------------------------------------------------------

                           EDIT

                            >-

------------------------------------------------------------
```

Enter the command "NEWF" to indicate the desire to start a new edit file. The system will output on the display:

```
------------------------------------------------------------

                         ADDR L=

------------------------------------------------------------
```

This is asking for the beginning address of the edit file, lower byte first. Enter "000". The system will respond with:

```
-------------------------------------------------------

                    ADDR L=000 H=

-------------------------------------------------------
```

This is asking for the high byte of the beginning address. As you enter information, PEDIT places the information in memory locations above this starting point. Enter "025" indicating that you wish the editor to start the file on page 25. This is the beginning of lesson space in the PILOT interpreter and should be used as the beginning of all edit files that are not to be merged with other edit files. The system will respond with the message:

```
-------------------------------------------------------

                        INPUT

-------------------------------------------------------
```

This indicates it is ready for lesson input from the keyboard. The following lesson can be typed in at this point in the sequence:

```
   * START
  T: WELCOME TO PILOT
   : WHAT IS YOUR NAME?
  A: $NAME
  T: CONGRATULATIONS ON
   : PILOTING PILOT
   : LETS HOPE NEITHER
   : OF US CRASH!
  T: $NAME, WHEN YOU ARE
   : TYPING A RESPONSE
   : YOU CAN ERASE A
   : CHARACTER BY TYPING
   : A "DELETE" OR KILL
   : A LINE BY TYPING
   : CTL/U PROVIDED YOU
   : HAVE NOT TYPED "RETURN."
```

```
    : WHEN YOU PRESS RETURN
    : I WILL GO BACK TO
    : PHIMON
  A:
BYE:
```

For details on how to use editor, see Appendix G.A. To get out of the input mode of editor, enter an extra "carriage return." Next step is to determine the location of the end of the edit file. Enter the command "DEOF" which means display end of file. The system will respond with an address in the L= H= format used to input the edit file beginning. Note the page number that of the end of the file (H=). For the purpose of this sample, assume the display was H=27.

The next step is to enter the "escape" character. This will cause the editor menue to be displayed. Next enter choice three to return to PHIMON. If you create an edit file that is too large for the editor, you will be given an error message "MEM OVERFLOW." You must delete a part of the file before proceeding.

4.2 Save Lesson

The next step is to create a permanent file of the lesson. This is done by "saving" the lesson on tape. Assume that the lesson is going to be placed on the system tape. Enter the command:

```
SAVE MYLSN 25-27*5000
```

SAVE tells PHIMON what operation is desired. MYLSN is the name PHIMON is to use for the lesson. Twenty five is the starting page of the file. The value 27 is the ending page of the file (systems reponse to DEOF). The value 5000 is the place in memory that the editor starts executing. The format is:

```
  SAVE(blank)(filename)(blank)(file start)-(file end)*5000
```

File names can be one to seven alphanumeric characters.

4.3 Execute Lesson

The next step is to execute the lesson with the PILOT interpreter to insure it performs properly. Enter the command:

```
"RUN PILOT"
```

This will cause PHIMON to load a copy of the PILOT interpreter from page 0 through page 24 (see Figure G-1). PHIMON will then be executed. Enter name as requested by lesson. When "return" is pressed in response to the last A: statement, the interpreter will execute the bye statement thereby returning to PHIMON.

4.4 Build Student Tape

Place PILOT system tape in drive one. Enter the

```
         !----------------------------------------------
         !                            ^
         !                            !
         !                            !
         !                            !
         !                            !
         !                            !
         !        PILOT LESSON        !
         !                            !
         !                            !
         !                            !
         !                            !
         !                            !
         !                            !
         !                            V
Page 25  !----------------------------------------------
         !                                       ^
         !                                       !
         !     PEDIT or PILOT                    !
         !                                       !
         !       This space used for PEDIT       !
         !       program while editing lesson.   !
         !                                       !
         !                                       !
         !       This space used for PILOT       !
         !       interpreter while executing     !
         !       lesson.                         !
         !                                       V
   0000  !----------------------------------------------
```

MEMORY MAP

Figure G-1

command

```
SAVE#1 FIRST 0-27*2340
```

This command tells PHIMON to save the file on drive one with the name MYLSN, using memory pages zero through 27, with a starting address of 2340. The format is:

```
SAVE(#1)(blank)(filename)(blank)(0)-(end page)*2340
```

If the file were placed on the system tape (drive zero), the (#1) would not be used and a single blank would be placed between the word "SAVE" and the filename.

What has been created is a copy of the PILOT interpreter with a set of starting instructions. Recall that the PILOT interpreter must have starting instructions and cannot be run without a lesson loaded. The lesson provides the necessary starting instructions. For a student to use this lesson, he would merely have to place the tape on drive zero, press the "on" button, disable the ROM and then give the command "RUN FIRST."

To update the edit file, that was saved upon exiting the editor, the sequence is:

```
LOAD MYLSN
RUN PEDIT
```

When the menue appears, select two or three. When the edit prompt appears, enter the command "EDIT." The editor will ask for the file address, enter "000" and "025." The editor will respond by displaying the first line of the edit file. Changes can be made as desired. Saving procedure is the same as for a new file. The last step prior to leaving editor is enter the command, "DEOF" to obtain the end page for the file saving process.

APPENDIX G.A

PILOT EDITOR COMMANDS AND FILES OPERATIONS

1. PURPOSE

Lessons are created and modified by the PILOT EDITOR. Lessons are stored on digital cassette tape through the use of the PHIMON operating system. The purpose of this Appendix is to describe the commands of the editor and the necessary operations of PHIMON. It is not intended to be a comprehensive document for PHIMON operations.

2. FILES OPERATIONS AT STARTUP

If an exising file is to be edited, it must be loaded prior to starting the PILOT editor. Assume that an existing filename, "OLDLSN," contained a lesson to be updated. Further assume that the tape cassette containing this file was a system tape on drive zero. The command to load the file would be "LOAD OLDLSN." If the tape were on drive 1, the command would be: "LOAD#1 OLDLSN."

This is only necessary when an existing file is to be edited. The next step is the same for either editing an existing file or creating an original file. Enter the command: "RUN PEDIT." Once it is loaded and executing, the editor will display a menue

```
-----------------------------------------------------------

                  PILOT Z-80
                    EDITOR

1.  EDIT PILOT LESSON - DISPLAY

2.  EDIT PILOT LESSON - HARD COPY

3.  RETURN TO PHIMON

          MAKE SELECTION PLEASE

-----------------------------------------------------------
```

Selecting choice one sets editor to output to the display only. Selecting choice two sets editor to output to the hard copy device and the display. If choice two is made, insure the printer is turned on first. If the printer is not turned on, the system will hang when choice two is selected. If this occurs, press the reset button (blue button next to the red ON/OFF switch) and enter the command "START" to reenter the editor menue.

3. INITILIZING COMMANDS

Once a selection is made, the display outputs a prompt:

```
------------------------------------------------------------

                           EDIT

                            >-

------------------------------------------------------------
```

If an original file is to be created, enter the command "NEWF." If an existing file is to be edited, enter the command "EDIT." This step must be done each time the editor is entered from the editor menue. In either case, the editor will respond with the prompt:

```
------------------------------------------------------------

                       ADDR   L=

------------------------------------------------------------
```

It is requesting the starting address for the edit file, low byte then high byte. Unless you are creating a file that is to be merged with another file (very rare operation), respond with "000" and "025". This is the lesson start location for PILOT and will be used almost always. If the command "NEWF" was entered, the editor will respond with "INPUT" and be ready for lesson entry. If the command "EDIT" was entered, the editor will respond by displaying the first line or first page as specified by the output mode (see mode command below).

4. EDIT COMMANDS

These commands are used to display and edit the edit file (block quoted materials below are from reference 5). All edit commands work with respect to the current line. In a majority of instances, the current line is the last line displayed. The software maintains a pointer to the current line at all times. The term "string," as used in this discussion, means any sequence or valid ASCII characters, less functions (such as carriage return).

COMMANDS	FUNCTIONS
A string	Add the string to the end of the current line and display the result.
BOTM	Place the current line pointer to the end of file.
C %stringl%string2	Find the first occurrence of stringl in the current line and change it to string2. The two string lengths need not be equal and the second string can be null (i.e., a C/R following the second delimiter). The delimiters (%) may be any printing ASCII character.
D (M)	Delete the current line (or M lines beginning with the current line) from the file. The file is moved in memory so that no empty space exists in the file. M is input as a decimal number - maximum value = 255. Value in parentheses is optional. If used, do not use parentheses.
F string	Find and display the first line in the file which begins with the string. The search begins with the line following the current line and continues until a match is found or the EOF is reached. The found line becomes the current line.

I string

Insert the string as a new line following the current line. If no string is included or if only a C/R is input as a command, the editor enters the continuous input mode. In this mode, multiple lines may be entered in the file by typing in each line followed by a C/R. Exit from the continuous input mode is accomplished by inputting a null line (C/R only). When the continuous mode is entered, the message INPUT will be displayed. Upon exiting this mode, the message EDIT will be displayed. No prompt is issued between multiple input lines which indicates that the editor is in the input mode.

INSM M

Insert M lines from memory following the current line (M = 1 to 255). The file is moved in memory to accommodate the new lines. The location (starting address) of the new lines will be requested and must be input from the keyboard. This command is designed for merging together of the two files but may also be used to move lines within the same file if the destination is at a higher memory address than the source. If this is not the case, only one line at a time may be moved correctly within the file.

LIST

List the entire file on the output device.

L string

Locate and display the first line in the file which contains the string anywhere within the line. The search begins with the line following the current line and continues until a match is found or the EOF is reached. The located line becomes the current line.

N (M)

Move the current line pointer to the next line in the file (or move M lines) and display the new current line. M may be positive or negative (max. range = ± 255). The value in parentheses is optional. If used, do not include parentheses.

P (M)

Print the current line (or M lines).

The last line printed becomes the new current line. The value in parentheses is optional.

PAGE — List one page (15 lines) beginning with the current line. The current line is unchanged.

R string — Replace the current line with the input string and display the result.

T — Set the current line pointer to the top of the file and display the first line or page of the file.

5. UTILITY COMMANDS

The utility commands allow displaying of the various pointers used by the program and specifying parameters to the program. All addresses output by these commands are displayed in split octal, low order address first followed by the high order address. The utility commands interface with the display and do not output to the hardcopy device.

COMMANDS	FUNCTIONS
DISP	Displays current line pointer. This command is useful for the INSM command to determine the starting address of the lines to be inserted.
DEOF	Display end of file address.
DISM	Display current setting of maximum memory size.
SETM	Set maximum memory address. This command requests an address input.
MODE L,P	Sets the output to the line (L) or page (P) mode. In the line mode, only the current line is displayed following a command. In the page mode, 15 lines are displayed. The first line displayed is the current line.
RUBO X	Sets the rubout character to X. X (initilalized to }) may be any printing ASCII character. The rubout character erases the previous input character in a command line. Multiple rubouts may be used to erase (back up) multiple characters.
KILL X	Sets the kill charcter to X. X (initialized to {) may be any printing ASCII character. The kill character deletes the entire input line. If the kill and rubout are set to the same character, the kill function will take precedence.

ESCAPE	Exit to PEDIT menue.
Q	Quit. Exit to PEDIT menue.

6. ERROR MESSAGES

The program will output the error message "WHAT?" in response to unrecognizable or improperly formatted commands. In addition to this general error message, several other error messages may be displayed.

On all commands which require an address input, the address is tested against the minimum useable file address. If the input address is less than the minimum, the error message "MIN ADDR (LH) = XXX YYY" will be displayed. This prevents overwriting of the editor program by the file being edited. The minimum octal address is L = 000; H = 025.

If a command is entered which increases the size of the file, the new end of file location is tested against the set maximum memory value. If the maximum would be exceeded by the command, the message "MEM OVERFLOW" is displayed and execution of the command is inhibited.

During execution of the INSM command, the data to be inserted is verified to be valid ASCII data. (Note: ASCII data, as defined in this program, is the 64 character upper case subset.) If a non-ASCII character other than a control character recognized by the program is encountered, the message "BAD DATA XXX YYY" is displayed where XXX YYY is the address of the invalid data. Execution of the INSM command is terminated if this error is displayed.

If a command such as Print M causes the end of file to become the current line, message "BOTTOM" will be displayed. This message will also be displayed if a Find of Locate command fails to match the input string indicating that the string is not present in the portion of the file searched.

7. TERMINAL FILE OPERATIONS

The user must determine the last page of the editor file before leaving the editor. Enter the command DEOF (display end of file). The editor will respond with an address L byte and H byte. Note the H byte. Press the "escape" key to exit the editor operating mode. This will place the editor in the Editor menue. Enter three to return to PHIMON.

The next operation is to save the edit file on tape with the PHIMON file command SAVE. PHIMON file names can be one to seven alphanumeric characters. Assume the file name "MYLSN" is desired and the DEOF command high address element was 56. The save command should be entered as follows:

```
"SAVE MYLSN 25-56*5000."
```

The format is:

```
SAVE(blank)(filename)(blank)25-(end page)*5000.
```

The number 25 is the starting page of the edit file and the "*5000" indicates the starting address of PEDIT. The principle variable items are the filename and ending page. If the tape to receive the file is on drive one, the terms "#1" is added directly behind the word save as follows:

```
SAVE#1(blank)(filename)...
```

This completes the editor instructions. The user is cautioned to initialize the edit file each time entry is made from the edit menue. If this caution is not observed, the entire edit file can be lost.

APPENDIX H

PILOT STUDENT SYSTEM USERS MANUAL

Table of Contents

1. PURPOSE

The purpose of this manual is to provide instruction on how to use the PILOT student system. Using PILOT as a student requires no knowledge of computers. This manual will provide all the information needed. The first step in taking a lesson is to obtain the lesson tape from an instructor. This tape is a cassette recording tape and contains all the necessary information for the computer.

2. STARTING THE COMPUTER

The computer has two switches that will be used. The first of these is the ON/OFF switch. It is a large red push button located at the lower right corner of the computer cabinet. This switch is on when the switch is lighted. The other switch is called a "ROM Disable Switch." The location of this switch varies between computers so ask one of the computer support personnel to point out its location. Make sure the on and off position of this switch is understood.

To use the system, place the tape in the tape drive number provided on the tape cassette with the lesson name on the top side. Place the ROM Disable Switch in the off position. Press the on button. The button should light and stay lighted. The tape should begin to move. After a few moments, the computer display will contain the message "RD OVLY" and then "RD DIR." Soon after these messages appear, the display will contain a list of one or more file names. Check to insure the lesson you wish to take is among the lesson names on the display.

If the system does not work as described above, the reason is probable incorrect position of the "ROM Disable Switch." In this case, turn the power off (by pressing the red ON/OFF switch again). Then move the ROM Disable Switch to the opposite position and start over by pressing the red ON/OFF switch. If the system still does not operate as described above, ask for help from the computer support personnel.

3. TAKING LESSONS

The process of taking lessons is one of the computer displaying information then the student typing in the the answers to questions. When you are typing into the computer, your answer is ended by pressing the key marked "RETURN." If you wish to change an answer, you can do so until the "RETURN" is pressed. To delete a character, press the "DELETE" key. Each time the "DELETE" key is pressed, one character will be deleted. To delete a whole line, press the "CNTL" and "U" key at the same time. The character or line that has been deleted will not be erased from the screen but will be ignored by the computer.

At some points in the lesson, the computer may have to get more information from the tape. If this occurs, the screen will remain unchanged while the computer moves the tapes. This operation will take a minute or two, so do not be concerned over no change in the display. At the very end of a lesson, the display will clear and the message "<PHIMON>" will appear on the screen. This means that session is over. If you wish to start a new lesson, just enter "RUN," a "blank," and the lesson name you desire.

4. ERROR MESSAGES

When the computer expects you to enter a numerical answer and you enter an answer containing non-numeric characters, it will give you the message:

"NUMERIC ENTRY REQUIRED."

You will be given another chance to enter a number.

Almost any other message that comes out of the system is a message that is the result of an error in the manner the lesson was prepared by the instructor. It would help the instructors greatly if, when an error does occur, you would copy down the two lines on the display above the error message and the error message itself. The following is a complete list of error messages:

ERROR MESSAGE	ACTION FOR STUDENT
*NUMERIC ENTRY REQUIRED	REENTER ANSWER
*NO ROOM	NOTE AND CONTINUE
*LABEL NOT FOUND	NOTE AND ASK FOR HELP
*USE DEPTH EXCEEDED	NOTE AND ASK FOR HELP
*ILLEGAL EXPRESSION	NOTE AND ASK FOR HELP
*VALUE > 99	NOTE AND CONTINE
*VALUE < -99	NOTE AND CONTINUE
*PILOT LESSON DESIRED NOT ON THIS TAPE. ASK FOR HELP	NOTE AND ASK FOR HELP

5. STOPPING THE COMPUTER

At the end of the lesson, the student may stop the computer by turning off the power. CAUTION: Before turning off the power, make sure none of the tapes are moving. To turn off the power, press the red ON/OFF button. You may remove the tape you place in the system by pressing the black tape at the right, front of the cassette drive.

APPENDIX I

REFERENCES

[1] Rubin, Sylvan. "A Simple Instructional Language," Computer Decisions, Nov. 1973, pp. 17-18.

[2] Starkweather, John A. "Guide to 8080 PILOT, Version 1.1," Dr. Dobb's Journal of Computer Calisthenics and Orthodontia, April, 1977 Vol. 2, No. 3, pp. 17-29.

[3] "Flyer Number 12," The Digital Group Inc, P.O. Box 6528, Denver, Colorado 80206.

[4] Starkweather, John A. "Source Code For 8080 PILOT, Version 1.1," Dr. Dobb's Journal of Computer Calisthenics and Orthodontia, May, 1977 Vol. 2, No. 5, pp. 17-34.

[5] Greeb, F. J. "A Classy 8080 Text Editor," Dr. Dobb's Journal of Computer Calisthenics and Orthodontia, June/July, 1976 Vol. 1, No. 6, pp. 13-26.

[6] Morgan, David J. "A Line-Oriented PILOT Editor," Dr. Dobb's Journal of Computer Calisthenics and Orthodontia, April, 1976 Vol. 3, No. 4, pp. 22-25.

A PILOT LANGUAGE SYSTEM

by

LARRY TRISTAN WALKER

B. S., Georgia Institute of Technology, 1961

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1978

PILOT has the distinction of being one of the simplest Computer Assisted Instruction (CAI) languages. Its simplicity is a major strength. Users rapidly become effective at creating PILOT programs. Students execute these programs with little difficulty. The simplicity of the language makes it possible to implement an effective PILOT system on a small, microprocessor computer system.

This report describes the implementation of a PILOT system on a Zilog Z-80 microprocessor system. The system makes maximum use of existing software modules. The PILOT system provides support to authors who create PILOT programs. These programs are referred to as CAI lessons. It provides support to students who will take these CAI lessons.

The principle modules of the system are a PILOT lesson editor for creation and changing lessons and a PILOT interpreter for executing the PILOT lesson.

This report outlines the creation of the software modules used as sources in this project and the modifications required to integrate these modules into a system. Users manuals for authors and students are provided. Lessons written in the PILOT programming language are also included to permit users to become familiar with the system by taking PILOT lessons.