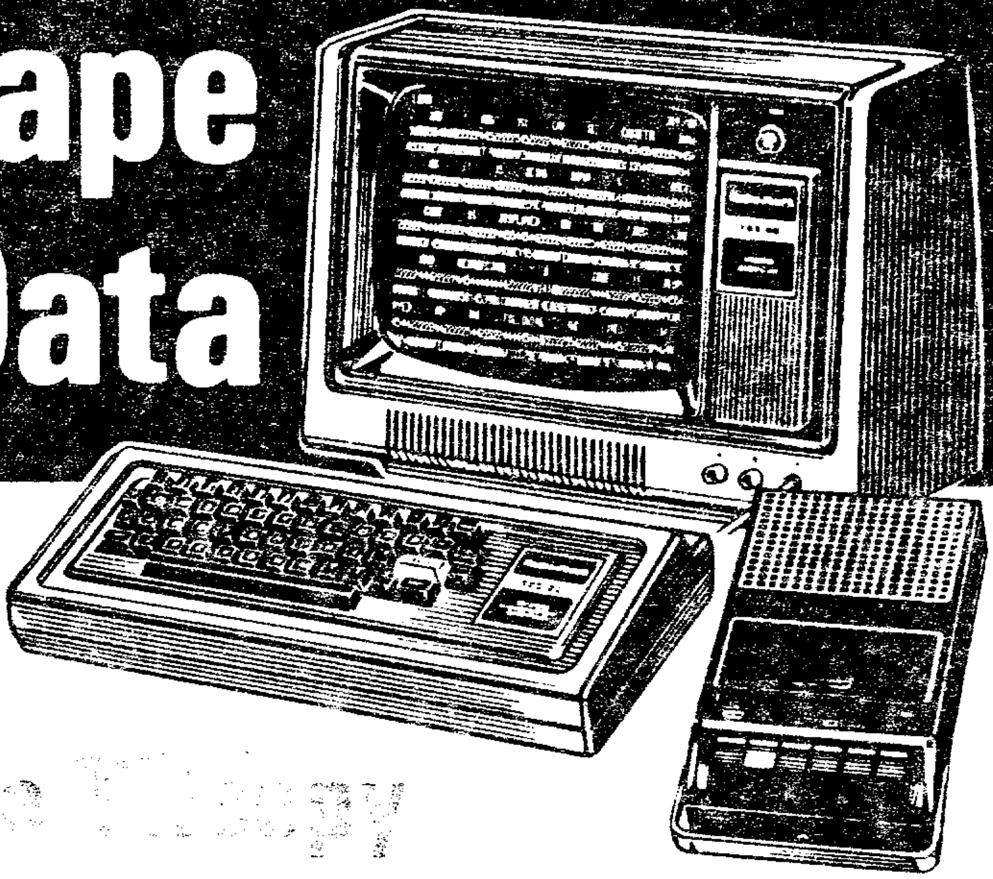
See and Copy Tape Data



use TRcopy

WITH YOUR LEVEL II TRS-80*

*TRS-80 IS A TRADEMARK OF THE TANDY CORPORATION

TRcopy

CASSETTE COPY

CO[illegible]

TRcopy

CASSETTE TAPE
COPY AND VERIFY SYSTEM

Published by

Data/Print PUBLISHING

P.O. BOX 903 • FARGO, N. D. 58107

CONTENTS

APPENDIX

INTRODUCTION

The use of cassette tapes to store digital data provides two big advantages over other forms of data storage. Cassette storage is low in cost and it is re-usable.

However, cassette tape storage does have some disadvantages and one of them is its narrow margin of tolerance for errors. Add to this the fact that you cannot see the data on the tape, and the results can mean time-consuming problems.

There are many possible mechanical and human errors that can occur while attempting to input data from a cassette tape. All of these errors will have the same result though--the tape will not load correctly.

When a tape does not load, the ordinary courses of action are very few and very inadequate.

This is where the TRcopy system comes in. By providing a means to display the contents of the tape on the CRT screen, many other corrective measures can be taken. Then after the problem is corrected, the TRcopy system can be used to prepare a new tape and to verify the correctness of the new tape.

HOW THE TRcopy SYSTEM WORKS

The TRcopy System is a machine language program that allows you to load any binary data from a TRS-80 Level II cassette tape into memory storage. The type of data does not matter because the TRcopy system handles all data in the same way.

The TRcopy system program resides in the lowest area of user RAM memory and the tape data is loaded into memory immediately following the copy system program. By using this format, the TRcopy system can be used with any size memory from 4k to 48k. The memory size will, however, govern the maximum length of a block of tape data that can be loaded or copied. The following chart lists the maximum block lengths for each of the available memory sizes:

Memory Size	Maximum Length of Tape			Maximum Count On CTR-41 Recorder
	Bytes	Minutes	Seconds	
4k	1,782	0 -	30	19
16k	14,070	3 -	58	154
32k	30,454	8 -	35	335
48k	46,838	13 -	12	516

For example, if you have a 16k machine, the maximum length of a program block that can be loaded is 3 minutes and 58 seconds. This would be approximately equal to a count of 154 on a CTR-41 tape recorder. The maximum byte count would be 14,070 bytes. The byte count is displayed on the screen and includes all tape codes as well as base data.

The data is always loaded into the same section of memory and held there for further processing. The data is loaded in an un-executable form. That is, even though a program is loaded in its entirety, it cannot be executed.

The data is loaded exactly as it is input from the cassette, which includes all tape processing codes.

As the data is being input from the cassette, it is displayed on the CRT in hexadecimal form with ASCII representation where applicable. The data is displayed at the same speed as it is input from the cassette.

Up to 320 bytes of data are displayed on the CRT screen at one time. If the data being input is longer than 320 bytes, the original data is overlaid on the screen until the second set of 320 bytes is displayed. This routine is continued until all of the data is input from the cassette tape.

As the data is being input and displayed on the CRT screen, it is stored in memory in its entirety. It is not overlaid in memory as it is on the CRT screen.

As each byte is input, it is counted and the total count is displayed in the upper right hand corner of the screen. When all of the data has been input from the tape, this count shows the total number of bytes used for memory storage of the data.

After a block of tape data has been loaded, the data may be checked for accuracy through the TRcopy Verification System. This is done by rewinding the tape and allowing the same data to be input in the verification mode. In this mode, each byte of data from the tape is matched against its corresponding byte in memory.

As the data is being verified, it is also displayed in hexadecimal code similar to the way it was displayed when it was loaded. If a byte of data from the tape does not match its corresponding byte in memory, an asterisk (*) is displayed over that byte on the CRT screen.

At the end of the verification period, a notation, "GOOD" will be displayed if all bytes from the tape match their corresponding bytes in memory. If any data does not match, the notation, "BAD" will be displayed.

When the data has been loaded and verified, you may want to make printed copies for closer analysis or future reference. If your TRS-80 system includes a line printer, the entire contents of the tape can be output to the printer in the same format that was used for the screen display.

The last function of the TRcopy system is to provide tape copies of the data as it is stored in memory. This data is output exactly as it was input including all processing codes. If the data was input as a BASIC program, it will be output as a BASIC program. If it was input as a machine language program, it will be output as a machine language program.

Any number of copies may be made while the data is stored in memory. The data does not change in memory until another tape is loaded.

The verification system may be used with tape copies just as it is used with the original tape. The principle is that if the original tape is verified with memory, and the tape copies are verified with the same memory, the tape copies must, therefore, be exact copies of the original.

TRcopy DEMONSTRATION

For this demonstration you will need a blank cassette tape and a program tape that you wish to copy to the blank cassette. The program tape for the demonstration should be one that is not especially difficult to load.

The program can be recorded either in BASIC or in machine language. It would be best to use a short program for which you know the approximate ending location of the data on the tape.

If you do not know the ending location of the data, set the cassette counter to zero and load the tape into your TRS-80 computer. When the tape has stopped, make a note of the counter reading.

The TRcopy program may now be loaded as follows:

1. Ready Level II Basic.

2. Load the TRcopy cassette and push Play. (This must be recorder Number 1 if you have more than one recorder on an expansion interface.)

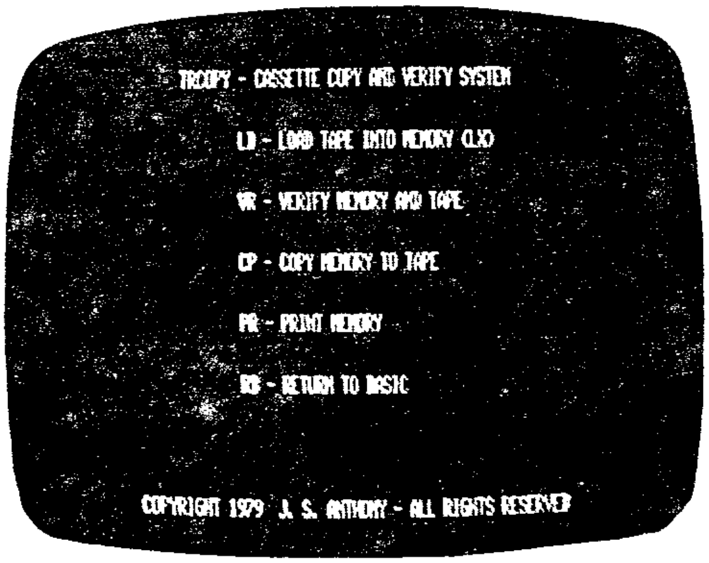
3. Set the volume to 5 or 6.

4. Set the tone to high.

5. Type "SYSTEM" and Enter.
   An asterisk and question mark (*?) will appear.

6. Type "TRCOPY" and Enter.
   The tape reader will start and the two asterisks in the top right hand corner will flash. After the program has loaded correctly, the tape will stop and another asterisk and question mark (*?) will appear.

7. Type a slash (/) and Enter.
   The following message will appear on the screen:



```
TRCOPY - CASSETTE COPY AND VERIFY SYSTEM

LD - LOAD TAPE INTO MEMORY (LX)

VR - VERIFY MEMORY AND TAPE

CP - COPY MEMORY TO TAPE

PR - PRINT MEMORY

RB - RETURN TO BASIC

COPYRIGHT 1979 J. S. ANTHONY - ALL RIGHTS RESERVED
```

8. Rewind the TRcopy tape and remove. (You cannot copy the TRcopy tape with the TRcopy system. If you attempt to do this, the system will return to BASIC.)

The TRcopy cassette is a high-quality tape that has been recorded under the most exacting conditions. Each tape has passed several rigid tests before being okayed for shipment.

In spite of these precautions it is still possible that a tape may be difficult to load.

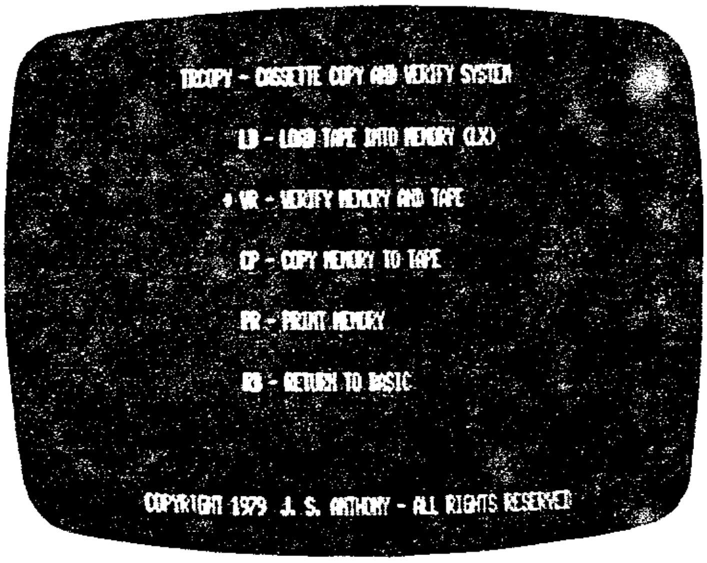
If the TRcopy tape does not load into your machine the first time, try using a slightly higher or lower volume setting. If this proves effective, write this setting on the label of the TRcopy tape for future reference.

If you still have difficulty, try turning the tape over. TRcopy is recorded on the other side for backup purposes directly opposite the first recording.

Instructions for the general care and use of cassette tape recorders is contained in the Appendix at the back of this manual. We cannot stress too much, the importance of these common housekeeping rules because there is no way to load accurate data from a recorder that is not functioning properly.

TRcopy MENU SELECTIONS

There are five selections on the menu index. Each selection is coded with a 2-letter abbreviation. When you type the first letter of the abbreviation, an arrow will appear to the left of the selection. For example, if the letter "V" is typed, an arrow will appear to the left of the verify selection as follows:



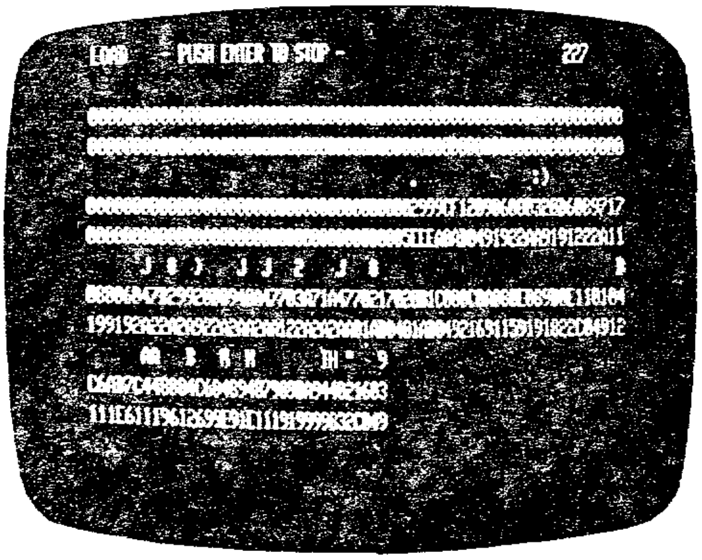
```
TRCOPY - CASSETTE COPY AND VERIFY SYSTEM

      LD - LOAD TAPE INTO MEMORY (LX)

    → VR - VERIFY MEMORY AND TAPE

      CP - COPY MEMORY TO TAPE

      PR - PRINT MEMORY

      RB - RETURN TO BASIC


COPYRIGHT 1979  J. S. ANTHONY - ALL RIGHTS RESERVED
```

If the second letter of the abbreviation is typed, the system will branch to that selection. If a letter other than the second letter is typed, the arrow will be removed and the system will return and await the input of another first letter.

LOADING TAPE DATA WITH "LX" INSTRUCTION

Now insert the program tape that you wish to copy into the cassette recorder. (Be sure that the tape is rewound to the beginning.)

Then push the Play key on the cassette recorder. The program is now ready to load. When the tape does start running, there will be a delay of from a few seconds to more than 30 seconds while the tape leader is being read. Then a set of double zeros will begin to spread across the screen and finally actual data from the tape will be displayed. The screen will appear similar to the following:



```
LOAD  - PUSH ENTER TO STOP -                227
```

Now type the letters "LX" and when the data has filled one half to three fourths of the screen, push the Stop key on the tape recorder.

LEADERS AND SYNC BYTE

Notice that the first two to four sets of data contain zeros. These zeros represent leaders that must preceed every program or block of data on tape.

At the end of these leaders, the data is displayed in hexadecimal format. On the line over the hexadecimal code, the intermittent characters displayed are the ASCII representation of the hexadecimal code. For example, a hexadecimal "41" represents the ASCII letter "A". A complete list of ASCII codes is contained in the Appendix.

Some hexadecimal codes do not represent ASCII characters and will therefore remain blank.

Data is written to magnetic tape in groups of eight bits using very precise timing cycles. There is no special starting or stopping code to set apart each byte of data. In order for the computer to read a magnetic tape, it must start reading during the zero leaders and test every group of eight bits for a predetermined sync byte code.

The sync byte is a code that the computer uses to get "in step" with the tape and it is located at the end of the zero leaders. The sync byte code for the TRS-80 is a hexadecimal A5 or binary 10100101.

The routine that is used to test for the sync byte is used only once for each block of data. Once the sync byte routine is entered, the computer will continue looking until it finds a sync byte or until the routine is aborted.

When the computer has detected an A5 sync byte, it must be allowed to continue reading through to the end of the block of data at the exact baud rate for which it was created.

If the tape reader is interrupted for any reason, the data will be out of sync and reading must be restarted with the zero leaders preceeding the sync byte.

Normally when you input a program, the system immediately goes into the sync test routine but the TRcopy system provides a separate instruction to bypass the sync test so that you can see exactly what the computer is reading. This is the "LX" instruction that you used to load the demonstration tape.

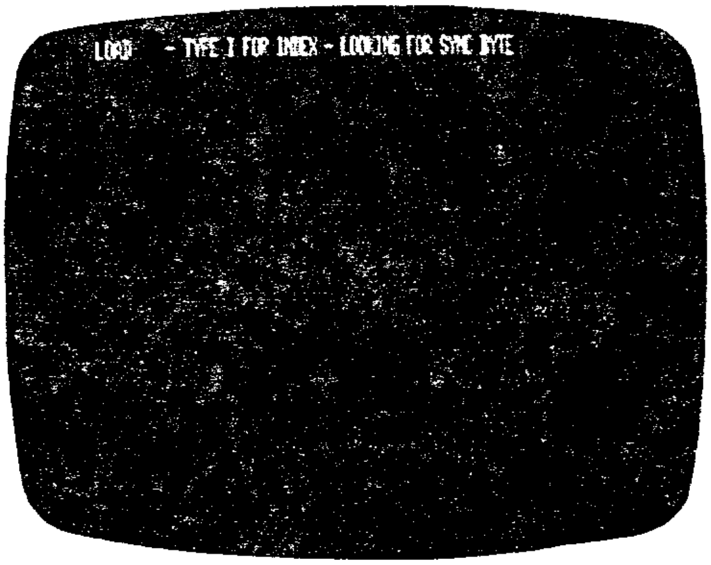
LOADING TAPE DATA WITH THE "LD" INSTRUCTION

Now back to our display on the CRT screen. When the TRcopy system input this data, it did not test for a sync byte. There is however, one chance in eight that the data could still be in sync without using the sync test routine.

In most cases, you will want to use the sync test routine and we will now demonstrate how this is done. First push the Enter key then type the letter "I". This will return you to the menu index. Now rewind the cassette and push the Play key.

This time we will use the sync test routine which is done automatically with the "LD" instruction. When the sync test routine is used, the zero leaders will not be displayed or stored in memory. The display will start with the A5 sync byte and proceed with actual data from the tape.

While the sync test routine is being executed, the screen will be cleared and the message "Looking For Sync Byte" will be displayed. This will last for a period of time depending upon the amount of blank tape and zero leader data that is contained before the sync byte.



```
LOAD   - TYPE I FOR INDEX - LOOKING FOR SYNC BYTE
```

Now type the letters "LD" and when the data has filled one half to three fourths of the screen, push the Stop key on the tape recorder.

TAPE DATA HEADERS

Unless the program on your tape was expremely short, you probably stopped the tape before the end of the data block. If you try to start the tape at this point, the data would continue from where it stopped but it will be out of sync and it would not load properly.

In order to load the program properly, the tape must again be rewound and the sync search routine again executed. When this is done, your tape data will probably fill the screen and start again by overlaying the previous data. Before doing this, there are several things that you should observe with regard to the tape heading.

In all cases, the first hexadecimal code should be an A5. This does not correspond to an ASCII character, so the first set of data should appear as follows:

Line 1- (blank)
Line 2- A
Line 3- 5

From here on, the format of the data will depend upon the kind of data that was input. The four standard formats for tape data used in a Level II TRS-80 are as follows:

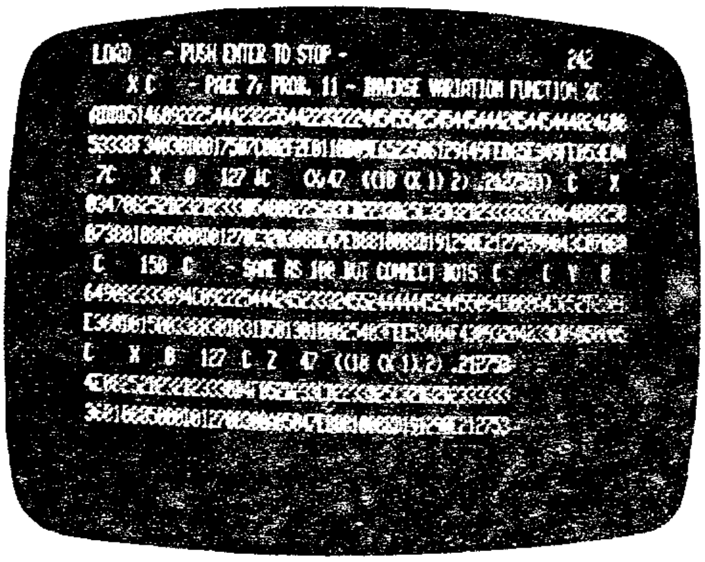
1. BASIC program format.
2. Machine language program format.
3. Data file created from a BASIC program
4. Assembler Language listing.

The detail of each of the tape formats is contained in the Appendix. For this demonstration, we will consider only the two most likely formats which are the BASIC format and the machine language format. A thorough understanding of these formats is not necessary in order to use the TRcopy system. However, it is helpful to be able to recognize each of the formats in order to identify the type of data being displayed and to find the program or file name.

Your demonstration tape should contain one of the following two headers depending upon whether it is a BASIC program or a machine language program:

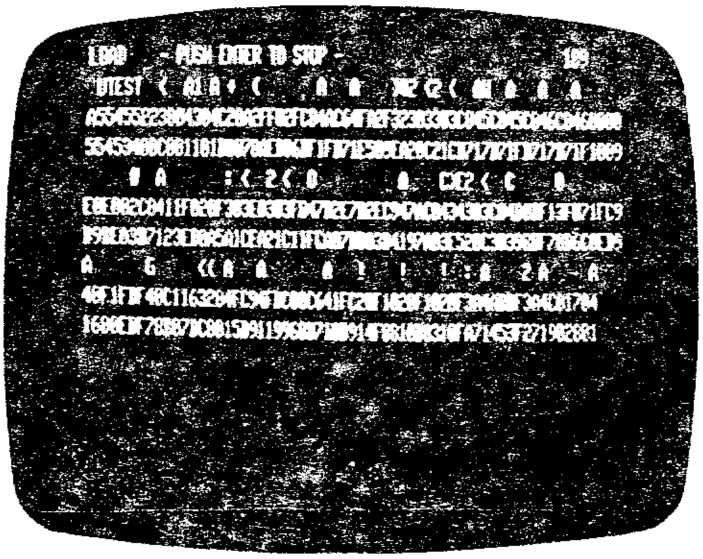
Header for BASIC program:		Header for Machine Language Program:	
A5	Sync Byte	A5	Sync Byte
D3	BASIC Start Codes	55	Machine Language
D3	(No ASCII		Start Code
D3	representation)		(ASCII letter "U")
58	Program Name	54	
	(58 is ASCII "X")	45	6-Byte Program Name
		53	(This example is
		54	TESTƀƀ.)
		20	
		20	

The program name for the BASIC program in the example is "X". The program name for the machine language program is "TEST". Unused bytes in the machine language program name are always "padded" with blanks which appear as hexadecimal code 20 for an ASCII space.



```
LOAD   - PUSH ENTER TO STOP -                 242
   X C   - PAGE 7, PROB. 11 - INVERSE VARIATION FUNCTION
```

Sample of BASIC header



```
LOAD   - PUSH ENTER TO STOP -                 189
 DTEST
```

Sample of machine language header

LOADING DATA BLOCK

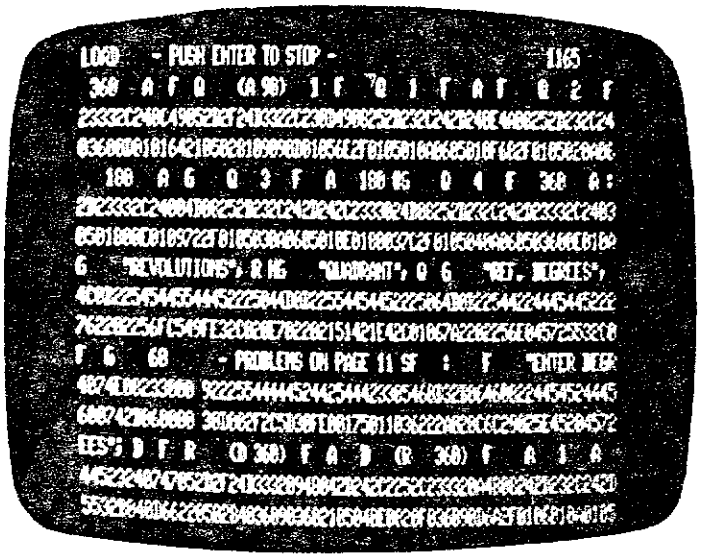
Now we will proceed with the demonstration by loading the entire program from the tape. First, push the Enter key and then type the letter "I", which will return the system to the index. Now rewind the cassette and push the Play key. The tape counter should also be set to zero at this time.

If the tape contains more than 320 bytes of data, the original data will be overlaid until the entire program is input. Unless there is another program or block of data after the program you are using, the display will end abruptly but the computer will continue reading the tape.

If another block of data is positioned after the one that was loaded, the leaders for the second block will begin to appear. The reader can be stopped any time within the leaders of the second block. This is the reason that we suggested you use a tape for which you know the ending location of the data. When the counter nears the end of the data, you should begin watching for new leaders or an end to the display.

Now type the letters "LD" and when the entire block of data has been loaded, push the Stop key on the tape recorder. The Enter key must then be pressed to stop the load function. After the Enter key is pressed, you have two options: First, you can use the Backspace key to erase a byte at a time from the screen and from memory; or second, you can type the letter "I" and return to the index.

The Backspace key may be used to erase data that is not part of the block that you are loading. This would generally consist of leaders from a second block of data. These leaders do not have to be erased as they will have no effect on the loading of duplicate copies. Erasing data should be done with caution because if even one byte of the main block is erased, copies of this block will not be usable.



```
LOAD   - PUSH ENTER TO STOP -                1165
```

At the top right hand corner of the screen, the total number of bytes required for the data is displayed. In the example, 1165 bytes were used. Make a note of the size of the data block. You will find it helpful during copying and verification to know the approximate location of the reader in relation to the beginning or end of the data. Note also that this count is not decremented when trailing zeros are deleted but the new count will be used for copy and verification purposes.

The data you loaded will be stored in memory until you use the load function again. After the sync byte has been found during the load function, the previous data is lost and cannot be recovered unless it is reloaded. The sync test routine can be aborted by typing the letter "I" while the "Looking For Sync Byte" message is being displayed.

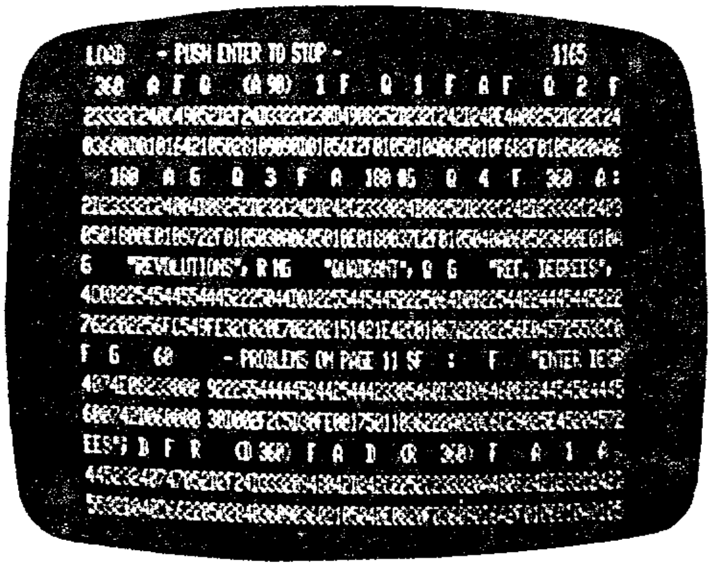
If you want to load a data block that is not in first position on the tape, just advance the tape to the leader area of that block and use the "LD" function as you did on the previous demonstration.

TAPE BLOCK TRAILING DATA

Now notice the last three bytes of data in the program. The last three bytes of significant data should appear as follows--depending upon whether the program is in BASIC or in machine language:

Last three bytes of
a BASIC program:

00	End-of-line code for last line in the program.
00 00	End-of-program code for a BASIC program.



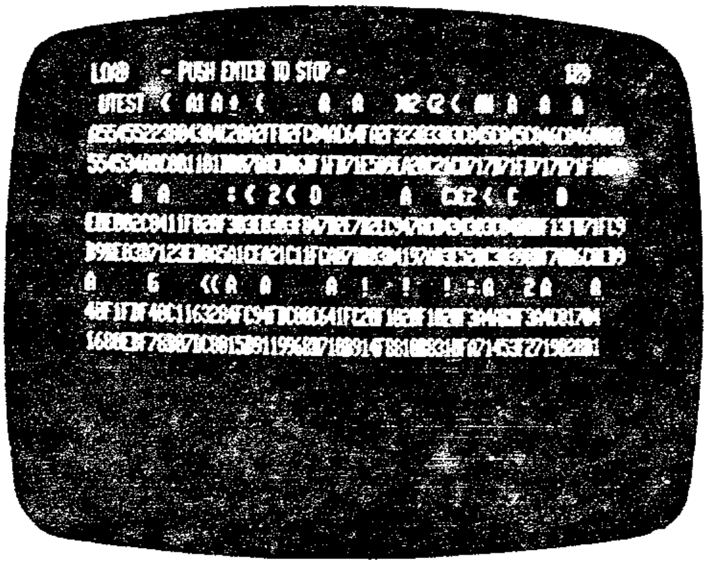
```
LOAD   - PUSH ENTER TO STOP -                    1165
```

Every BASIC program must end with three sets of hexadecimal zeros. The first set is actually the end-of-line code for the last line in the program. The last two bytes are simply hexadecimal zeros indistinguishable from ordinary leaders.

Last three bytes of a machine language program:

78 End-of-program code for a machine language program.

00
41 Address to begin program execution with the least significant byte first. The program in this example begins execution at address hex 4100.

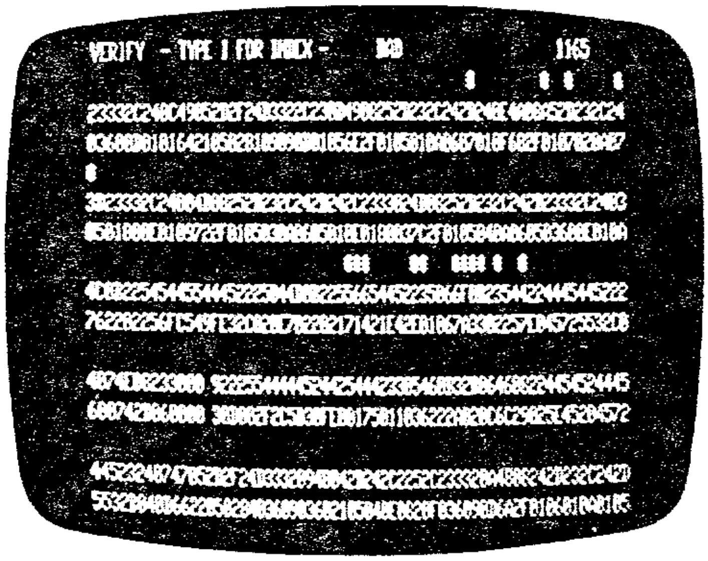


```
LOAD    - PUSH ENTER TO STOP -
```

Every machine language program must have a hexadecimal 78 in the third from last position. This is the end-of-program code for a machine language program. The last two bytes of a machine language program may contain the address that execution begins. After you have loaded a machine language program with a SYSTEM instruction, you would normally type a slash (/) and then the Enter key. This causes the system to branch to this address to begin execution. If the beginning address is not provided in this way, or if you wish to begin execution somewhere other than this location, you may enter a decimal address after the slash. In that case the system will branch to the address that you enter.

VERIFYING MEMORY WITH TAPE CONTENTS

Your demonstration program is now contained in memory but before further processing, you should always verify memory with the tape contents. The verification process compares every byte of data on the tape with its corresponding byte of data in memory. The data will be displayed on the screen as it was when it was loaded except that the first line will not contain ASCII characters. The first line will contain an asterisk (*) for any byte of data on the tape that does not match its corresponding byte in memory.



```
VERIFY - TYPE I FOR INDEX -   BAD          1165
```

At the end of the verification function, the message, "GOOD" will appear if no asterisks were displayed. If any asterisks were displayed, the message, "BAD" will appear.

You may now proceed to verify the memory with the tape contents. Type the letter "I" to return to the index. Rewind the tape and push the Play key on the cassette reader. Now type "VR" and the system will begin to read the tape.

The screen will be cleared and will remain so until significant data is encountered at the end of the zero leaders.

When the data has been verified, rewind the tape and push the letter "I" to return to the index.

You may repeat the verification process as often as you want. When you have successfully verified the tape with memory contents, you may want to try an experiment with the tape recorder volume control. To do this, start the verification process again. While the tape is being verified, gradually increase the volume control until the asterisks begin to appear.

At this point it is very likely that the data will be thrown out of sync as evidenced by the presence of an asterisk over every byte. You will have witnessed the effect that an overly high volume setting has on the read process. This experiment does not effect the tape nor does it change the memory contents. If you re-adjust the volume to a normal setting, you will find the tape can be verified as before.

The above experiment can also be done by adjusting the volume control downward. When the minimum setting is reached, the data will stop being input or displayed but the reader will continue running. If the volume is then increased, the data will be out of sync and an asterisk will be displayed over every byte.

From this experiment, you can determine the minimum and maximum settings for a particular cassette tape. You may want to try this on both the original tape and the tape copies. You should find that the tape copies provide a wider range of tolerance for the volume setting. This experiment also provides revealing results when used with different brands or qualities of cassette tapes.

COPYING MEMORY TO TAPE

Now that the data has been loaded and verified, you may make duplicate tapes of the original. Making tape copies is very simple. First load a blank cassette and push the Play/ Record keys. Be sure the tape is rewound and positioned on a recordable section of tape. Then type the letters "CP".

The recorder will start and the screen will be cleared and stay clear while the leaders are being output. After the leaders have been output, the data will be output to the tape recorder and to the CRT screen at the same time. The data will appear in hexadecimal code as it did during loading and verification but the ASCII characters will not be displayed.

When the data has been completely duplicated, the display and the tape recorder will stop. Then push the Stop key on the tape recorder and rewind the tape. Type the letter "I" to return the system to the index.

VERIFYING TAPE COPIES

The tape copies should be verified just as the original was verified. Load the tape copy and push the Play key on the recorder. Then type the letters "VR".

When the tape recorder stops and the message "GOOD" appears, you will know that the duplicate is a perfect copy of the original program. The tape may be rewound and the system returned to the index by typing the letter "I".

At this time you may want to try the <u>input</u> volume setting experiment on the tape copy to see how it compares in tolerance with the original tape. This can be done by repeating the verification process then increasing or decreasing the volume setting until the asterisks begin to appear.

The volume setting has little or no effect on the <u>output</u> of data to cassette tapes because the CTR-41 has an automatic level control. The recording level can, however, provide for additional experimentation with the volume control during verification. Try recording with the volume control set as high as it will go then check the tape for tolerance levels using the verification function. The tolerances should be approximately the same regardless of how high or low the setting was when it was recorded.

WRITING MEMORY TO PRINTER

The last function of the TRcopy system is to output the contents of memory to a printer. If your TRS-80 system includes a line printer or a quick printer, you may find it useful to print out the contents of the cassette tape. With a printout, you will be able to reference the complete program as opposed to a maximum of 320 bytes on the CRT screen. A printout also provides for closer analysis of checksums and other tape codes, which is somewhat difficult to do on the CRT screen.

Sending the memory contents to a printer is very simple. First line up the paper then turn on the printer. Be sure all switches are on.

Now type ‘‘PR’’. The printer will advance one line and proceed to print the contents of memory in the same format as used on the screen display. The first line of each set contains the ASCII code (if applicable) and lines two and three contain the hexadecimal code. Each line will contain 64 characters. The data will continue printing without interruption until all data that was loaded has been printed.

After the data has been printed, the system will return to the index. If you want to stop the printer and return to the index at any time, type the letter ‘‘I’’. This is also true if you should happen to enable the Print function when a printer is not connected or turned on. (The system does not lock up like LPRINT!)

IN CONCLUSION

Any of the preceeding four functions may be repeated in any order as often as necessary. When you are ready to cease operations, type "RB" and the system will return to Level II BASIC.

We hope that you now have a better overall understanding of cassette tape storage systems. The TRcopy system is not a difficult system to use but it is important to have an understanding of tape data formatting in order to proceed effectively.

If you are not totally familiar with the system, we would suggest you go back and review from the beginning. By going through the demonstration a second time, you should have no difficulty in applying the system to your needs.

A condensed copy of the instructions is contained in the Appendix. These simple instructions should provide you with a fast, easily understood guide for everyday use of the TRcopy system.

TAPE FORMAT FOR BASIC PROGRAM

Data block header for BASIC program:

A5 Sync Byte

D3
D3 BASIC Start Codes
D3

58 Program name in ASCII. The name in the example is "X". (Refer to illustration below.)

Line format in a BASIC program:

1F Line Pointer. This is the address of the last byte in the program line.
43 The least significant byte is first. The address of the last byte of the line in this example is hex 431F.

64 The line number in hexadecimal. The least significant byte is first. The
00 line number in the example is hex 0064 or decimal 100.

93 The code for the BASIC instruction. All Level II BASIC instructions are converted and stored as single-byte hex codes. The number in the example is the code for the REM statement. See pages 31, 32 and 33 in the Appendix for a complete list of TRS-80 BASIC codes.

20 Spacer. Hexadecimal code for ASCII blank.

2D
20
} Operators or constant data of varying lengths.
4F
4E

00 End-of-line code, always represented by a hexadecimal zero.

Data block end code for BASIC program:

00 End-of-data code for a BASIC program, always
00 represented by two hexadecimal zero bytes.

Note: There is no checksum in a BASIC program.

EXAMPLE OF FIRST AND LAST SECTIONS OF A BASIC PROGRAM TAPE FORMAT

```
    % C      - PAGE 7, PROB. 11 - INVERSE VARIATION FUNCTION %C
ADDD5146092225444232255442233222445455425454454444245445444024608
53338F34030D0017507C002F2E0110D09E6523506129149FE065E349FE053E04
```

```
G     "REVOLUTIONS", R NG     "QUADRANT", Q  G     "REF  DEGREES".
4C0B22545445544452225044D0B2255445445222506 4D0B2254422444544522 2
762202256FC549FE32C020E702202151421E42C01067A2202256E045725522

F  G    60
4074E08233000
600742D060000
```

TAPE FORMAT FOR MACHINE LANGUAGE PROGRAM

Data block header for machine language program:

A5 Sync Byte

55 Start Code

54
45 Program name in six ASCII bytes. The name in the example is "TEST".
53 (Refer to the illustration below.) Unused bytes are padded with blanks.
54
20
20

Internal block format for a machine language program:

3C Block header code.

80 Length of the block from hex 1 to 00, or decimal 1 to 256. (Hex 00 = decimal 256.) The length of the data block in this example is hex 80 or decimal 128.

00 Starting memory for the block with least significant byte first. The start-
41 ing memory for this block is hex 4100.

31
00 Data of varying lengths up to 256 bytes
} as defined by block length byte.
10
17

6D Checksum. The total of starting memory and all data for this block. The checksum total does not include the block header code, the block length byte or the checksum itself. The checksum is derived by adding all of the data into a single byte with any carries being disregarded.

Data block end for a machine language program:

78 End Code.

00 Address to begin execution with the least significant byte first. The
41 address where execution begins in this example is hex 4100. After the program is loaded, the system will branch to this address when you type a slash (/) and Enter. If you want to begin execution somewhere other than this location, you may enter a decimal address after the slash and the system will branch to that address.

```
 UTEST   <   A1  A  +   <           A    A      >*2 <2  <    AW   A_   A    A
A55455223804304C20A2FF02FC84AC64FA2F3223033030C845C845C846C8460000
55453400C001101DB078AED06DF1FD71E509EA20C21CD717D71FD717D71F1009

      #  A            . <  2 <  Q             A      <>E2 <   C       G
E0E002C8411F020F303E0303F847B2F7B2EC947AC84343030C84D0DF13F071FC9
B9BE03D7123ED0A5A1CEA21C11FCA07D603D4197A03E520C303986F70B6C0ED9

A        G         << A    A         A             !       !        A     2 A       A
40F1FDF40C1163284FC94FDC00C641FC20F1020F1020F3A4A6DF3A4C01704
16B0EBF78B07DC8015D911996BD710B914F8810B6310FA71453F271902B01
```

TAPE FORMAT FOR DATA FILE CREATED FROM A LEVEL II BASIC PROGRAM

Data block header for data file:

A5 Sync Byte

Numeric field format for a data file:

20 Sign of the numeric field. If the field is positive, this byte will contain a 20. If the field is negative, this byte will contain a 2D.

33
30 ASCII representation for a decimal
} number. Varying lengths.
34

20 End of numeric field code.

Field separator for a data file:

2C This is the hex code for an ASCII comma (,). A comma must separate every field in the data file.

Alphameric field format for a data file:

53
54
} ASCII representation for alphameric data of varying lengths.
44

Data block end code for a data file:

0D This is the ASCII representation for a carriage return.

Note: There is no checksum in a Level II data file.

```
  30 , 20
A2332223320
50300C0200D
```

The above is an example of a file created from a Level II BASIC program. The file contains two numeric fields. The first number is +30 and the second number is +20.

TAPE FORMAT FOR AN ASSEMBLER LISTING

Data block header for an assembler listing:

A5 Sync Byte

D3 Start Code

54
45 Program name in six ASCII bytes. The name in the example is "TEST".
53 (Refer to the illustration below.) Unused bytes are padded with blanks.
54
20
20

Line format for an assembler listing:

B0
B0 ASCII representation of the decimal line number in the assembler listing.
B1 Up to five digits may be used. Unused leading digits are filled with zeros.
B0 The line number in this example is a decimal 100.
B0

20 Spacer.

54 ASCII representation of the assembler labels, operation codes, operands
42 and comments, all of varying lengths. A horizontal tab is represented
} by hex 09.

0D End of assembler line. (An ASCII carriage return.)

Data block end code for assembler listing:

1A End Code.

Note: There is no checksum in an assembler listing.

EXAMPLE OF FIRST AND LAST SECTIONS OF AN ASSEMBLER LISTING

```
00100 TBUG      EQU    4380H  {  01000               OUT    (255),A
00110 VIDEO     EQU    3C00H  {  01010               LD     (STATUS),A
00120 KEY       EQU    2BH    {  01020               RET
00130           ORG    4100H  {  01030 STATUS        DEFB   0
00140 START     LD     SP,$   }  01040               END    START
```

EXAMPLE OF FIRST AND LAST SECTIONS OF ASSEMBLER LISTING TAPE FORMAT

```
  TEST          TBUG EQU 4380H          VIDEO EQU 3C00H          KEY EQU
AD545522BBBBB2545404550333340BBBBB25444404550343340BBBBB24450455
5345340000100042579515943808D0011006945F951593C008D001200B599515
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

         END START
0BBBBB2044405545501
D01040095E4934124DA
```

CONVERSION TABLE – 1

BINARY	OCTAL	DEC	HEX	ASCII	TRS-80 CONTROLS
0000 0000	000	000	00	NUL	
0000 0001	001	001	01	SOH	
0000 0010	002	002	02	STX	
0000 0011	003	003	03	ETX	
0000 0100	004	004	04	EOT	
0000 0101	005	005	05	ENQ	
0000 0110	006	006	06	ACK	
0000 0111	007	007	07	BEL	
0000 1000	010	008	08	BS	BACKSPACE
0000 1001	011	009	09	HT	
0000 1010	012	010	0A	LF	LINE FEED
0000 1011	013	011	0B	VT	VERTICAL TAB
0000 1100	014	012	0C	FF	FORM FEED
0000 1101	015	013	0D	CR	CARRIAGE RETURN
0000 1110	016	014	0E	SO	CURSOR ON
0000 1111	017	015	0F	SI	CURSOR OFF
0001 0000	020	016	10	DLE	
0001 0001	021	017	11	DC1	
0001 0010	022	018	12	DC2	
0001 0011	023	019	13	DC3	
0001 0100	024	020	14	DC4	
0001 0101	025	021	15	NAK	
0001 0110	026	022	16	SYN	
0001 0111	027	023	17	ETB	CONV TO 32 CH
0001 1000	030	024	18	CAN	BACKSP CURSOR
0001 1001	031	025	19	EM	ADVANCE CURSOR
0001 1010	032	026	1A	SUB	CRT DOWN LINE
0001 1011	033	027	1B	ESC	CRT UP LINE
0001 1100	034	028	1C	FS	HOME CURSOR
0001 1101	035	029	1D	GS	80 CH OR BEG LN
0001 1110	036	030	1E	RS	40 CH OR ERS LN
0001 1111	037	031	1F	US	20 CH OR CLR FR
0010 0000	040	032	20	SP	(SPACE)
0010 0001	041	033	21	!	
0010 0010	042	034	22	"	
0010 0011	043	035	23	#	
0010 0100	044	036	24	$	
0010 0101	045	037	25	%	
0010 0110	046	038	26	&	
0010 0111	047	039	27	'	
0010 1000	050	040	28	(	
0010 1001	051	041	29	)	
0010 1010	052	042	2A	*	
0010 1011	053	043	2B	+	
0010 1100	054	044	2C	,	
0010 1101	055	045	2D	-	
0010 1110	056	046	2E	.	
0010 1111	057	047	2F	/	
0011 0000	060	048	30	0	
0011 0001	061	049	31	1	
0011 0010	062	050	32	2	
0011 0011	063	051	33	3	
0011 0100	064	052	34	4	
0011 0101	065	053	35	5	
0011 0110	066	054	36	6	
0011 0111	067	055	37	7	
0011 1000	070	056	38	8	
0011 1001	071	057	39	9	
0011 1010	072	058	3A	:	
0011 1011	073	059	3B	;	
0011 1100	074	060	3C	<	
0011 1101	075	061	3D	=	
0011 1110	076	062	3E	>	
0011 1111	077	063	3F	?	

CONVERSION TABLE – 2

BINARY	OCTAL	DEC	HEX	ASCII	TRS-80 BASIC
0100 0000	100	064	40	@	
0100 0001	101	065	41	A	
0100 0010	102	066	42	B	
0100 0011	103	067	43	C	
0100 0100	104	068	44	D	
0100 0101	105	069	45	E	
0100 0110	106	070	46	F	
0100 0111	107	071	47	G	
0100 1000	110	072	48	H	
0100 1001	111	073	49	I	
0100 1010	112	074	4A	J	
0100 1011	113	075	4B	K	
0100 1100	114	076	4C	L	
0100 1101	115	077	4D	M	
0100 1110	116	078	4E	N	
0100 1111	117	079	4F	O	
0101 0000	120	080	50	P	
0101 0001	121	081	51	Q	
0101 0010	122	082	52	R	
0101 0011	123	083	53	S	
0101 0100	124	084	54	T	
0101 0101	125	085	55	U	
0101 0110	126	086	56	V	
0101 0111	127	087	57	W	
0101 1000	130	088	58	X	
0101 1001	131	089	59	Y	
0101 1010	132	090	5A	Z	
0101 1011	133	091	5B	↑	
0101 1100	134	092	5C	↓	
0101 1101	135	093	5D	←	
0101 1110	136	094	5E	→	
0101 1111	137	095	5F	_	
0110 0000	140	096	60		@ IN PRINT@
0110 0001	141	097	61	a	
0110 0010	142	098	62	b	
0110 0011	143	099	63	c	
0110 0100	144	100	64	d	
0110 0101	145	101	65	e	
0110 0110	146	102	66	f	
0110 0111	147	103	67	g	
0110 1000	150	104	68	h	
0110 1001	151	105	69	i	
0110 1010	152	106	6A	j	
0110 1011	153	107	6B	k	
0110 1100	154	108	6C	l	
0110 1101	155	109	6D	m	
0110 1110	156	110	6E	n	
0110 1111	157	111	6F	o	
0111 0000	160	112	70	p	
0111 0001	161	113	71	q	
0111 0010	162	114	72	r	
0111 0011	163	115	73	s	
0111 0100	164	116	74	t	
0111 0101	165	117	75	u	
0111 0110	166	118	76	v	
0111 0111	167	119	77	w	
0111 1000	170	120	78	x	
0111 1001	171	121	79	y	
0111 1010	172	122	7A	z	
0111 1011	173	123	7B		
0111 1100	174	124	7C		
0111 1101	175	125	7D		
0111 1110	176	126	7E		
0111 1111	177	127	7F	DEL	

CONVERSION TABLE – 3

				TRS-80	
BINARY	OCTAL	DEC	HEX	GRAPH	BASIC
1000 0000	200	128	80		END
1000 0001	201	129	81	🬀	FOR
1000 0010	202	130	82	🬁	RESET
1000 0011	203	131	83	🬂	SET
1000 0100	204	132	84	🬃	CLS
1000 0101	205	133	85	🬄	CMD
1000 0110	206	134	86	🬅	RANDOM
1000 0111	207	135	87	🬆	NEXT
1000 1000	210	136	88	🬇	DATA
1000 1001	211	137	89	🬈	INPUT
1000 1010	212	138	8A	🬉	DIM
1000 1011	213	139	8B	🬊	READ
1000 1100	214	140	8C	🬋	LET
1000 1101	215	141	8D	🬌	GOTO
1000 1110	216	142	8E	🬍	RUN
1000 1111	217	143	8F	🬎	IF
1001 0000	220	144	90	🬏	RESTORE
1001 0001	221	145	91	🬐	GOSUB
1001 0010	222	146	92	🬑	RETURN
1001 0011	223	147	93	🬒	REM
1001 0100	224	148	94	🬓	STOP
1001 0101	225	149	95	▌	ELSE
1001 0110	226	150	96	🬔	TRON
1001 0111	227	151	97	🬕	TROFF
1001 1000	230	152	98	🬖	DEFSTR
1001 1001	231	153	99	🬗	DEFINT
1001 1010	232	154	9A	🬘	DEFSNG
1001 1011	233	155	9B	🬙	DEFDBL
1001 1100	234	156	9C	🬚	LINE
1001 1101	235	157	9D	🬛	EDIT
1001 1110	236	158	9E	🬜	ERROR
1001 1111	237	159	9F	🬝	RESUME
1010 0000	240	160	A0	🬞	OUT
1010 0001	241	161	A1	🬟	ON
1010 0010	242	162	A2	🬠	OPEN
1010 0011	243	163	A3	🬡	FIELD
1010 0100	244	164	A4	🬢	GET
1010 0101	245	165	A5	🬣	PUT
1010 0110	246	166	A6	🬤	CLOSE
1010 0111	247	167	A7	🬥	LOAD
1010 1000	250	168	A8	🬦	MERGE
1010 1001	251	169	A9	🬧	NAME
1010 1010	252	170	AA	▐	KILL
1010 1011	253	171	AB	🬨	LSET
1010 1100	254	172	AC	🬩	RSET
1010 1101	255	173	AD	🬪	SAVE
1010 1110	256	174	AE	🬫	SYSTEM
1010 1111	257	175	AF	🬬	LPRINT
1011 0000	260	176	B0	🬭	DEF
1011 0001	261	177	B1	🬮	POKE
1011 0010	262	178	B2	🬯	PRINT
1011 0011	263	179	B3	🬰	CONT
1011 0100	264	180	B4	🬱	LIST
1011 0101	265	181	B5	🬲	LLIST
1011 0110	266	182	B6	🬳	DELETE
1011 0111	267	183	B7	🬴	AUTO
1011 1000	270	184	B8	🬵	CLEAR
1011 1001	271	185	B9	🬶	CLOAD
1011 1010	272	186	BA	🬷	CSAVE
1011 1011	273	187	BB	🬸	NEW
1011 1100	274	188	BC	🬹	TAB(
1011 1101	275	189	BD	🬺	TO
1011 1110	276	190	BE	🬻	FN
1011 1111	277	191	BF	█	USING

CONVERSION TABLE – 4

				TRS-80	
BINARY	OCTAL	DEC	HEX	TAB	BASIC
1100 0000	300	192	C0	0	VARPTR
1100 0001	301	193	C1	1	USR
1100 0010	302	194	C2	2	ERL
1100 0011	303	195	C3	3	ERR
1100 0100	304	196	C4	4	STRING$
1100 0101	305	197	C5	5	INSTR
1100 0110	306	198	C6	6	POINT
1100 0111	307	199	C7	7	TIME$
1100 1000	310	200	C8	8	MEM
1100 1001	311	201	C9	9	INKEY$
1100 1010	312	202	CA	10	THEN
1100 1011	313	203	CB	11	NOT
1100 1100	314	204	CC	12	STEP
1100 1101	315	205	CD	13	+
1100 1110	316	206	CE	14	-
1100 1111	317	207	CF	15	*
1101 0000	320	208	D0	16	/
1101 0001	321	209	D1	17	^
1101 0010	322	210	D2	18	AND
1101 0011	323	211	D3	19	OR
1101 0100	324	212	D4	20	>
1101 0101	325	213	D5	21	=
1101 0110	326	214	D6	22	<
1101 0111	327	215	D7	23	SGN
1101 1000	330	216	D8	24	INT
1101 1001	331	217	D9	25	ABS
1101 1010	332	218	DA	26	FRE
1101 1011	333	219	DB	27	INP
1101 1100	334	220	DC	28	POS
1101 1101	335	221	DD	29	SQR
1101 1110	336	222	DE	30	RND
1101 1111	337	223	DF	31	LOG
1110 0000	340	224	E0	32	EXP
1110 0001	341	225	E1	33	COS
1110 0010	342	226	E2	34	SIN
1110 0011	343	227	E3	35	TAN
1110 0100	344	228	E4	36	ATN
1110 0101	345	229	E5	37	PEEK
1110 0110	346	230	E6	38	CVI
1110 0111	347	231	E7	39	CVS
1110 1000	350	232	E8	40	CVD
1110 1001	351	233	E9	41	EOF
1110 1010	352	234	EA	42	LOC
1110 1011	353	235	EB	43	LOF
1110 1100	354	236	EC	44	MKI$
1110 1101	355	237	ED	45	MKS$
1110 1110	356	238	EE	46	MKD$
1110 1111	357	239	EF	47	CINT
1111 0000	360	240	F0	48	CSNG
1111 0001	361	241	F1	49	CDBL
1111 0010	362	242	F2	50	FIX
1111 0011	363	243	F3	51	LEN
1111 0100	364	244	F4	52	STR$
1111 0101	365	245	F5	53	VAL
1111 0110	366	246	F6	54	ASC
1111 0111	367	247	F7	55	CHR$
1111 1000	370	248	F8	56	LEFT$
1111 1001	371	249	F9	57	RIGHT$
1111 1010	372	250	FA	58	MID$
1111 1011	373	251	FB	59	' (REM)
1111 1100	374	252	FC	60	
1111 1101	375	253	FD	61	
1111 1110	376	254	FE	62	
1111 1111	377	255	FF	63	

INSTRUCTIONS ON THE
CARE AND USE OF CASSETTE RECORDERS

If you have more than occasional problems in loading cassette data, it could be that there is something mechanically wrong with your recorder. This is very likely if your recorder has ever been dropped or if it has been serviced by unqualified personnel.

A good way to narrow down mechanical problems is to interchange your TRS-80 components with those of a friend or with the demonstrator system at your Radio Shack store. (But be sure your components contain some sort of identification so that you can tell them apart.)

By interchanging tape recorders to load the same tape, it can become quite obvious when a recorder is not functioning mechanically.

On the other hand, it can also become quite obvious when a cassette tape is defective. If the same tape cannot be loaded in either of two different machines, then the tape itself is most likely defective.

The read/write head on your cassette recorder should be cleaned and demagnetized after every four hours of use. Data is recorded on magnetic tape by exposing it to a magnetic field created by the write head. If the write head itself becomes magnetized, the tape can pick up unwanted signals and distort the data.

Tape head demagnetizers are low in cost and readily available. To use a demagnetizer, place the pole tip against the recording head and move it over the surface for a few seconds. Be sure the tip is protected so that it does not scratch the head. Then slowly draw the demagnetizer away from the head by 12 or more inches, and unplug the demagnetizer. It is the slow withdrawal of the demagnetizer that returns the tape head to an unmagnetized condition.

The tape head and drive wheels should also be cleaned with a special solvent for that purpose. Data signals on magnetic tapes depend upon very precise timing and if the drive wheels are not clean, the tape can slip and loose synchronization.

You may wonder what effect the starting and stopping of the reader has on the data signals. If your recorder is in proper mechanical condition, you should be able to use the start and stop keys on the recorder at any time without damage to the data signals--as long as the recorder is running.

The condition to avoid is when the power is stopped with the read/write head contacting the tape. In this case it is possible that data could be distorted from residual magnetism in the tape head. Always push the Stop key on the tape player before using remote devices that turn off the power to the machine.

COMMON PROBLEMS ENCOUNTERED IN THE USE OF CASSETTE TAPES

The most common problem with machine language tapes is that they will not load because the checksum derived from the data does not agree with the checksum recorded on the tape. The reason the checksums do not agree is because the data is not being input exactly as it was created.

Even when a machine language program is successfully loaded back into memory, the checksum may not catch all read errors. If one byte is lower than it should be and another byte is higher by the same amount, the checksum will be correct but the program in memory will be incorrect.

Problems with BASIC programs that are CLOADed back into memory may be much more subtle. BASIC programs do not use checksums to verify reader accuracy. They are simply "dumped" back into memory including any read errors that may occur. Even when all visible signs indicate that the load was successful, there is a very high probability that read errors can occur. This same problem also applies to assembler listings.

When these errors do occur, they usually turn up as bugs in a program that was originally bug-free. Another sign of read errors in BASIC and assembler programs is discovered when you try to list the program. The listing will usually appear correct at first but at some point it will start listing unintelligible data and may eventually crash out of the operating system entirely.

Many times these problems can be temporarily corrected by using a higher or lower volume setting on the cassette recorder. This is not usually a permanent solution because if the recording is that sensitive to volume, there will probably be no setting at which it will always function properly.

The best solution is to create a copy of the tape using a high quality blank cassette taking care to be sure the recorder is clean and in good operating condition. The copy will be much more reliable and the original can be kept as a backup.

With the TRcopy system, this situation can be handled much more effectively than it could be by simply copying the tape. The TRcopy technique for copying a tape that is over-sensitive to the volume setting is as follows:

First, the tape is loaded into memory using a setting of about six on the volume scale. Since the TRcopy system does not test checksums, it will load the tape exactly as it is read.

Next, the tape is verified byte-for-byte with the contents of memory. An asterisk (*) will be displayed over any byte of tape data that does not match its corresponding byte in memory. If the verification is bad, it will be indicated by as little as a single asterisk to as many as an asterisk over every byte.

If any asterisks appear, the volume setting should be increased or decreased and the tape should be loaded and verified again. If the asterisks increase in number, the volume should be set in the opposite direction and the tape re-loaded and verified.

Once it is determined whether the volume should be increased or decreased to improve the input, a satisfactory load can usually be accomplished by repeating the procedure.

When a tape is especially troublesome, it is often wise to verify it several times. If the data is input identically that often, you can be quite certain that it is correct.

After the original tape has been loaded and verified, one or more copies of the tape should be made and verified. You will find that the volume setting for the copies will provide a normal range of tolerance.

There are some cases in which a defective tape cannot be corrected. If the tape was created with a strong but defective signal, it can be loaded, verified and copied normally and still contain the original defects. Of course, neither the original nor the copies could be loaded correctly with normal load instructions in this case.

The TRcopy system does not allow for making manual changes in tape data because such changes would be extremely difficult and impractical. If a tape cannot be loaded or verified in spite of repeated attempts, the only alternative is to obtain another copy from the original source.

OTHER LOADING PROBLEMS

The previous problem is by far the most common and most difficult to deal with if it is severe. Most other problems are caused by the fact that you cannot see the data on the tape.

One problem may be that you are using the wrong file name. It is very possible to give a name to a program and then proceed to forget the name. With the TRcopy system this is no problem at all. By using the Load function, the exact name can be determined because you can see the name on the CRT screen.

Another problem is that you may not know the location of the data that you are attempting to load. It is possible that the tape was not completely rewound when new data was recorded over old data. In that case the computer will start reading with the original data and will be out of sync when the new data is encountered.

Location problems can also occur when more than one program is contained on a single tape. If the exact locations are not documented when the programs are created, it may not be easy to locate them again.

These problems can be easily overcome by using the TRcopy system to load and display the tape contents. In this manner the exact location of the programs can be established and the tape can be used as it is or it can be corrected using the copy function.

Problems can also occur if the program basis is not known. You may be trying to load a machine language program with a CLOAD instruction. Or, you may be trying to load a BASIC program with a SYSTEM instruction.

In either of these cases the program can be displayed using the TRcopy system, and the format of the data can be used to identify the type of program or file that is causing the problem.

TRcopy CONDENSED INSTRUCTIONS

1. Load TRcopy Program:
   Ready Level II BASIC.
   Load TRcopy cassette and push play.
   Set volume to 5 or 6.
   Set tone to high.
   Type "SYSTEM" and Enter.
   Type "TRCOPY" and Enter.
   After the tape has stopped, type a slash (/) and Enter.
   Rewind program tape and remove.

2. Load Memory from Tape:
   Load cassette to be copied and push Play.
   Type "LD" to load tape into memory.
   When data is loaded, push Stop key on tape recorder.
   Push Enter key to exit load routine.
   Push Backspace key to erase byte from memory.
   Type "I" to return to index.

3. Verify Memory with Tape:
   Rewind tape and push Play.
   Type "VR" to verify.
   Rewind cassette and remove.
   Type "I" to return to index.

4. Copy Memory to Tape:
   Load blank cassette and push Record/Play keys.
   Type "CP" to copy.
   When tape stops, release Record/Play keys.
   Rewind tape.
   Type "I" to return to index.

5. Verify New Tape with Memory.
   (Same as No. 3 above.)

6. Print Contents of Memory:
   Line up paper and turn on printer.
   Type "PR" to output memory to printer.
   System will return to index when finished.
   Type "I" to cancel printing and return to index.

7. Any of the above 5 operations may be repeated in
   any order.

8. Cancel TRcopy
   Type "RB" to return to BASIC.

When the message "Looking for Sync Byte" is being displayed, the system will return to the index if the letter "I" is typed.