

DISK INDEXING PROGRAM VERSION 3.0 FOR THE TRS-80 MODEL I AND MODEL III  
Mumford Micro Systems - Box 400 Summerland, California 93067  
Copyright (C) 1982 by Bryan Mumford - All rights reserved

### INTRODUCTION

When Mumford Micro first offered a disk indexing program for the TRS-80 in the summer of 1979, it was the first one on the market. It was a "first generation" program and in spite of its limitations, a review in the first issue of 80 Microcomputing called it "the best thing since sliced bread". After the Model III appeared on the scene, it became necessary to rewrite the original DISK INDEX to accomodate the new machine and make other improvements, and Version 2.0 was a significant improvement over Version 1.0. But as the TRS-80 market matures it becomes more complex. For a time there were a few "independent" operating systems which were all compatible with TRSDOS. But after the appearance of the Model III the independent system developers got more independent and started to produce operating systems that were totally incompatible with TRSDOS. This makes it very difficult for a program that reads directories. Everybodies directories are displayed and stored differently. Consequently, it became necessary for DISK INDEX to get more complex as well, and version 3.0 was written. It is written entirely in machine language and as a result runs much faster than earlier versions. The most significant improvement, however, is its ability to read a directory from just about any TRS-80 disk. It will automatically recognize and catalog directories from any operating system for either the Model I or Model III, and it will do this on both machines. That means you can catalog Model I disks on a Model III, and Model III disks on a Model I (if it has a double density controller installed). The only limitation is with CP/M, which is not a TRS-80 operating system. At this time, DISK INDEX 3.0 will not catalog CP/M disks.

The purpose of this program is to organize, alphabetize, catagorize, and computerize your program library. If you have very many disks you are probably well aware of the problem of remembering which disk contains which version of which program. INDEX will simplify that task by allowing you to create a master index of all your program titles. This index may be searched, sorted, printed on a line printer, and saved on disk for future reference and update. Programs may be added by reading disk directories directly, from previously stored data files, or by entering them manually. A 48K machine will hold over 2500 programs.

### LOADING THE PROGRAM

INDEX is supplied on a special disk. This disk is not "protected" in any way, but it conatins a special loader for people with only one disk drive. If you have two drives, put a TRSDOS compatible system disk in drive 0 and the distribution disk in drive 1. Copy INDEX/CMD from the distribution disk to your own system disk.

If you have only one drive, you will need to use the special loader to copy INDEX onto your own TRSDOS disk. This special loader is designed to work with the current version of TRSDOS. Using another operating systems may produce erratic results. Put a current TRSDOS system disk in drive 0 and hit RESET. After DOS READY is displayed, remove the system disk, insert our distribution disk, and hit RESET again. The disk will boot up and display a message indicating which version of TRSDOS it expects to be working with. If this is not the version you are using, you should start over with the proper version. Otherwise, the distribution disk will find INDEX/CMD, load it into memory, and prompt you to put your system disk back in. You may then follow the prompts to save the program onto your own disk.

While the distribution disk is designed to work only with TRSDOS, the INDEX program itself will run under any operating system you have (except CP/M). Once you have copied INDEX to a TRSDOS system disk, you may move to any other operating system you like.

### GETTING STARTED

To run INDEX from DOS READY, just type INDEX and hit <ENTER>. INDEX will load and display it's command screen. For those of you too impatient to read directions (like myself), we will take a quick run through basic operation of the program. Afterwards, each function will be described in greater detail.

At the bottom of the command screen you will see that there are no disks and no programs in the index. Obviously, the first thing to do is to build an index. This is menu selection B. Type B (no <ENTER> is required) and you will be asked for a drive number. Type 0. You will then be asked for a disk name. This may be any name you like up to 8 characters long. After specifying a name, hit <ENTER>. Drive 0 will start up and the directory will be briefly displayed and stored. When the command screen returns you will see that there is now one disk in the index and a few programs. To see these programs, type L to list the index. Type P to list by program. Each program read from drive 0 will be displayed along with the name you gave that disk. The last "program" will be the disk ID name stored in the directory when the disk was first formatted. When you get to the end of the list, hit <ENTER>. Try the L command again, but this time follow it with D to list by disk. You will see that the disk listing also shows you how many free grans there are and the type of disk it is. Hit <BREAK> to return to the command screen. Now hit S to sort the index. The screen will flicker very briefly (because the list is so short). Type L to list the index and then P to list it by program. You will now see that all of the programs are in alphabetical order, with the disk ID at the top of the list. Hit <BREAK> to return to the command screen.

### GROUND RULES

To begin with, there are some general features of the program that need to be covered. For one thing, all commands in INDEX are entered with only one keystroke. The <ENTER> key is only required for those situations where you are asked to specify several characters, such as the disk name, a file name, or a number. When you are asked for one of these entries, the request will be followed by several periods. The number of periods represents the maximum number of characters you may enter. You may enter fewer characters than this, but INDEX will not allow you to enter more.

The program remembers which function you last selected from the command screen. To select the same one again, you may hit <ENTER> instead of the letter. This feature is most useful when you are building an index.

At just about any point in the program you may return to the command screen by hitting <BREAK>.

Screen listings will be shown a page at a time. That is, no more than 15 lines will be displayed at a time. You will then be given a prompt to hit the space bar to see the next screen. If you do not want the display to stop after each screen, just hold the space bar down and the listing will be continuous. If you want to stop any given screen, just release the space bar. To stop the display instantaneously, anywhere in the program, hit the down-arrow key. To resume, hit the up-arrow.

While running this program it is not usually necessary to have a system disk in drive 0. In fact, you can put just about any disk imaginable in drive 0 and index it. But when you exit the program, or when you try to WRITE AN INDEX ON DISK or READ AN INDEX FROM DISK, you must have a system disk in drive 0. Furthermore, it must be the same system as was in drive 0 when you first loaded the program.

### BUILD AN INDEX

This function is the one that reads directories into the program. It is used, as the title suggests, to build your index. After typing B, you are asked for a drive number or the <ENTER> key. The use of the <ENTER> key will be described in a moment. After specifying the drive number to look at, you will be asked for a disk name. This name may be any name you like up to 8 characters long. Once the directory has been read in, this name will be referenced by every program on that disk, and will be displayed in other parts of the program as the disk on which those programs may be found. It is a good idea to give some thought to what you will use as disk names. You may want to use a name that is descriptive of the programs on that disk, such as "GAMES", "UTILITY", or "DATA1". If you have a very large number of disks, however, you may want to use a number like "00012" instead of a descriptive name. If the name you enter is already in the file, INDEX will tell you so and ask if you still want to use it. If you don't, you will be asked for another name. If you do use this name, every program currently in the index that is assigned to that disk name will be deleted from the index before the new disk is read in. This makes it easy to keep your index current by allowing you to read in disks on which changes have been made without getting double entries on programs that haven't changed.

After specifying the disk name, hit <ENTER>. The directory on that disk will be found, displayed, and stored. You will then be returned to the command screen. If the directory was very small, it won't be displayed very long. This is not a problem though because INDEX has better ways to look at directories. When you sit down to build an index you will want it to go as fast as possible.

As mentioned earlier, when you first enter this function you are asked for a drive number or the <ENTER> key. If you enter a number, INDEX will try to get a directory from the drive you specify. It will search for many different types of disks, however, and it tries to identify them one after another until it finds the format you are using. If the disk it is looking at is one of the first formats it tries, it won't take very long. But if it is one of the last ones, it will take longer. If, for example, you have a Model III and are cataloging Model I double density disks, it first looks for Model III disks (of several types), then Model I single density disks (of several types), and finally Model I double density disks before it finds a match.

The <ENTER> key is used to speed up this process by using default values. The first time you read a disk in, specify the drive with a number. INDEX will then try all formats to identify the disk type. When it finds the right format, it will store that information as the default values. The next time you select BUILD AN INDEX, if you hit <ENTER> instead of a drive number, the last format and drive number used will be assumed to be valid again. If the new disk is the same format as the last one, INDEX will go straight to the directory and read it very quickly. If you happen to put in a different format, however, you will get either a DISK ERROR or a DIRECTORY NOT FOUND message. In these cases you will need to try again and specify the drive number to get INDEX to try all formats.

The fact that INDEX remembers your last function selection and that it remembers defaults for disk formats makes it possible to go very fast when you are initially creating an index. When you put the first disk in, use the B key to select BUILD AN INDEX. Then use a number key to specify the drive number. Enter a disk name and hit <ENTER>. After the command screen returns, take out that disk, put a new disk of the same format in the same drive, and hit <ENTER> twice. The first <ENTER> is interpreted as the default function selection, which is BUILD AN INDEX. The second <ENTER> is interpreted as the default drive and disk type selection. You will then be asked for the new name, and the new directory will be read much faster than the last one. If you get over eager and hit <ENTER> too many times, the request for a disk name will be repeated. It won't accept nothing for an answer. In this way you can quickly build an index without paying much attention to what you're doing. Not paying attention to what you do may not always be advisable, but in a repetitive task like this, the easier it is the better. If you pay so little attention that you get out of sync, remember that the <BREAK> key will always return you to the command screen.

When INDEX reads a disk, it reads more than program names. For one thing, it also reads the number of free grans left on that disk. This piece of information is kept with the disk name and is displayed whenever you ask to see what is on that disk. What exactly is meant by a gran, however, varies from one operating system to another. Basically, a gran may be defined as the smallest unit of disk space a file can use. If you have a file that is one byte long, it will take one gran to store it. In TRSDOS on the Model I, which has 2 grans per track, this means 1/2 of a 2,560 byte track or 1,280 bytes. In TRSDOS on the Model III, which has 6 grans per track, this means 1/6 of a 4,608 byte track or 768 bytes. The point is, different operating systems divide their tracks into different numbers of grans, and the number of actual bytes left on a disk depends on the operating system it uses. If you have a Model III NEWDOS disk (2 grans per track) and a MODEL III TRSDOS disk (6 grans per track) that each have 10 grans left, the NEWDOS disk will have 3 times as much room left on it.

Another piece of information that is read from each directory is the type of disk that it is. There are basically three types of disks: Model I single density, Model I double density, and Model III. When you ask INDEX to list it's index by disk, in addition to showing you the program names and the number of free grans left on each disk, it will also display the type of disk that it is. This piece of information is also coded into the disk ID name. When the directory is read, the disk ID name is stored as a program. The disk ID is the name that was specified when the disk was first formatted. If it is a system disk, the disk ID is the name of the operating system. Since this name is often significant, INDEX reads and stores it with the programs. Since it is not a program, however, it is handled a little differently. For one thing, it is not included in the count of programs shown on the command screen. For another, it has two characters added before and after it. These characters cause the disk ID names to be alphabetized to the beginning of the index and also indicate which type of disk it is. For example, a TRSDOS Model I disk will have the ID name TRSDOS. INDEX will change that to -TRSDOS-. The single lines mean it is a single density Model I disk ID and not a program. A Model I DOUBLE DOS disk will have the ID name DBLDOS. INDEX will change that to =DBLDOS=. The single and double lines mean it is a Model I double density disk. A Model III TRSDOS disk will have an ID name TRSDOS, but INDEX will change it to ==TRSDOS==. The double lines indicate that it is a Model III disk.

When reading directories in this mode there are at least a couple of ways to run into problems. You could specify a drive that you don't have, you could forget to close the door on a drive that you do have, or you could fill your memory to capacity in the middle of a directory read. In any of these cases an error will be generated and the function terminated. Under these conditions, however, it is possible to end up with a disk in the file that doesn't have some or all of it's programs assigned to it. For this reason, if an error occurs while BUILDING AN INDEX, you should DELETE the last disk name you specified before the error occurred.

#### SORT THE INDEX

This command selection will alphabetize all the programs in the index. It uses a Shell-Metzner sort and is quite fast. If your index is small you will see the screen barely flicker as it sorts and returns to the command screen. If your index is larger, you will see the word "\*\*\* SORTING \*\*\*" and a series of periods as the sort progresses. You may abort this function with the <BREAK> key. When it has finished alphabetizing it will automatically return to the command screen.

#### LIST THE INDEX

This function is used to display and print the index. At the top of the screen will be a message indicating the printer status (on or off). If this message says that the printer is off, the index will only be shown on the video screen. If the message says the printer is on, the index will be sent to the line printer as well as the video screen. See PROGRAM VARIABLES for a thorough description of printer controls. INDEX supports two kinds of outputs: by program and by disk. The information is essentially the same in both cases, it is just arranged differently.

### PROGRAM LIST

If you request a program listing, every program in the index will be shown a video page at a time, followed by the name of the disk it may be found on. If the index has been alphabetized, the first "programs" will not be programs at all, but disk ID's and their type identifiers (see BUILD AN INDEX for an explanation of the disk ID's). If you do not want the display to pause after each screen, hold the space bar down.

If the printer is on, the display will also go to the line printer. If the printer is not at the top of a page when the listing starts, a form feed will be done. This is necessary because the the line printer output is formatted in two numbered columns which are designed to fit on whole pages. To set top of form, see PROGRAM VARIABLES. When the printer is on, the video display will change a little. Since the video screen isn't wide enough to show two columns, they will be compressed into alternate lines of a single column. In other words, the first program shown will be number 1. The second one will be number 55 (the first item in the second column). The third one will be number 2, etc. This will look a bit funny on the video screen but will look fine on paper. Each program displayed will be preceded by a number. You may notice, however, that the last program in the index has a number larger than the number of programs that are supposed to be in the index according to the command screen. The reason for this is that the program listing includes the disk ID names, which are not really programs. The command screen does not include disk ID names when counting programs.

When the printer is on, the display will not pause between video pages. It will, however, pause between line printer pages if you have asked it to (see PROGRAM VARIABLES), and it will skip enough blank lines at the bottom of each page to bring it to the top of the next page. Each page will also be titled and numbered. The listing may be halted at any time by hitting the down-arrow key. Hit the up-arrow key to resume. The listing may be aborted by hitting the <BREAK> key. When the list is finished you will be asked to hit <ENTER> to return to the command screen.

### DISK LIST

If you request the listing by disk instead of by program, the display will start with the first disk and show it's name, the number of free grans on it, and the disk type. The disk ID and all of the programs on that disk will then follow. The display will pause after each disk and ask you to hit the space bar. If the printer is on, these pauses will not occur. When printing on paper, INDEX will try to avoid putting half of a disk on one page and half on another, but if a directory is large this may happen anyway. The pages will be titled and numbered, however, and this should prevent confusion. The listing may be halted at any time by hitting the down-arrow key. Resume by hitting the up-arrow key. You can abort the listing by hitting <BREAK>. When the list is finished you will be asked to hit <ENTER> to return to the command screen.

### FIND AN ITEM

This function is used to find a specific program, disk, or type of program. Line printer output is supported in this mode, and a printer message will tell you if it is on or off. For a thorough discussion of printer controls, see PROGRAM VARIABLES. The first question you will be asked in this mode is whether you want to find a program or a disk. If you specify P for program, you will be asked for the characters you want to search for. You may enter up to 12 characters. The index will then be searched for any programs that contain these characters. In looking for a match, INDEX will look anywhere in the program name. The characters you specify don't have to be the first ones in the program. When a match is found, the program will be displayed on the video screen, along with the name of the disk it's on, as in the LIST mode. If the printer is on, they will also be printed on the line printer, though in this mode the printer output will be in only one column since INDEX can't know how many matches there will be until it finds them. If no matches are found you will be told so. You may also use the FIND mode to search for types of programs. For example, suppose you wanted a list of all of your BASIC programs. Also suppose that all of your BASIC programs end with the extension /BAS. If you specify /BAS as the characters you want to search for, all BASIC programs will be displayed.

If you say you want to find a disk rather than a program when you enter the FIND mode, you will be asked for the characters you want to search for. Since disk names have a maximum of 8 characters, this is the maximum number you can search for. The list will then be searched for any disk names that contain those characters. If a match is found, all of the programs on that disk will be displayed as in the LIST mode, and you will be asked to hit the space bar to continue. If the printer is on, this information will also be sent to the line printer, and the pause will not occur. If there are more disks that match the characters you are looking for, they will all be shown in the same manner. If no match is found you will be told so.

#### ADD AN ITEM

There may be times when you want to add a program to the file manually. If what you are adding is really a program, it is probably better to do it automatically by reading the disk into the BUILD AN INDEX function again. This will also correct the free grans value. You can, however, add a program manually, or you can add a word or phrase that is not a program but will describe the disk that you add it to. For example, if you have a very large index and every disk in it has a number for a name, you can add a 12 byte "program" which is really a description of that disk. If the first character in this "program" is an ASCII character of lesser value than a hyphen, these descriptions will be alphabetized to the first items in the index, where they can be easily examined. Two characters that precede the hyphen are the asterisk and the left parenthesis. Some disk descriptions that could be added this way, and would be alphabetized to the beginning of the index, are "\*GAMES DISK", " ACCOUNTING", and "(ASSEMBLER)".

To add a program, you will first be asked for the name of the disk you want the program to go on. If you enter a disk name that is not in the index you will be told so. If you enter a valid disk name, that disk will be found and all programs on it will be displayed. You will then be asked for the name of the program you want to add to it. This program will be added, and the disk will be displayed again with the addition. When you do not want to add any more programs you may hit either <ENTER> without specifying a program or <BREAK>. Both keys will return you to the command screen.

#### DELETE AN ITEM

This function will allow you to delete a single program or a whole disk. If you delete a single program, the number of free grans on that disk will be different than the number stored in INDEX, and for that reason it is better to read the whole directory over again with BUILD AN INDEX when changes are made on a disk. When you enter this function, the first thing you will be asked for is the name of the disk you want to delete from. If you enter a disk name that is not in the index you will be told so. If you enter a name that is in the index, it will be displayed along with all the programs that are on it. You will then be given three choices: you may delete a program, delete the entire disk, or skip the disk shown and look for another disk with the same name. It should be mentioned that you always have at least one other option, which is to hit the <BREAK> key and return to the command screen.

If you type the slash key and hit <ENTER>, INDEX will look for another disk name that matches the characters you specified. If one is found you will again be given the three options mentioned above. If no match is found you will be told so.

If you enter a program name, it will be deleted from the disk (note that the grans free will now be incorrect). If you enter a program name that is not on this disk you will be told so. If you hit <ENTER> without specifying any program name you will be returned to the command screen.

If you type an asterisk and hit <ENTER>, the entire disk will be deleted from the index and you will be returned to the command screen.

WRITE THE INDEX ON DISK

Let's say you just spent 30 minutes loading all of your disks into INDEX. You have 2000 programs stored, alphabetized, and perfectly organized. You will want some way to preserve this information for later use, and that's what this function is for. It will write a data file on disk that will contain the entire index (just the data, not the program). This data file may then be loaded back into INDEX some other time to be examined, printed, or updated. When you select this function you will be warned to have a system disk in drive 0. Be sure the system disk you use is the same system that you had in drive 0 when you loaded INDEX. Operating systems are like horses: you can't change them in mid-stream. Because of the way these data files are loaded back into the program, it is necessary that all files written to disk be in alphabetical order. Rather than making this your responsibility, INDEX takes the liberty of calling the alphabetizing function itself if the index is not currently alphabetized. It will then ask you for the name you want to give the data file. The file name may have up to 8 characters, a 3 character extension, and a drive specification (INDEXFIL/ASC:1). As the index is saved, the record numbers will be displayed on the video screen. You will be returned to the command screen when it is finished.

READ AN INDEX FROM DISK

This function will read back in an index that has been previously saved on disk with the preceeding function. As INDEX will warn you, be sure you have a system disk in drive 0 before using this function. As each program or disk name is read in, it's record number will be displayed on the screen. INDEX is clever enough to prevent you from loading a file that is not really an INDEX data file, and will tell you if you try to.

If there are already disks in the index when you use this function, they will not be erased. Because of this, one index may be appended to the end of another. INDEX is not clever enough, however, to know if you end up with the same disk entered twice this way. That will be your responsibility. If this situation occurs, INDEX will see them as separate disks. The fact that they are nearly identical is beside the point, because there are internal differences that INDEX sees and you can't. If you know that an appended index has the same disk in it twice, just DELETE one of them. Reading that disk in again with the BUILD AN INDEX function will only replace the first one.

Because of the way INDEX stores it's data, it will seem to slow down when appending a data file to an already large index. That's because it has to pack each new disk name under all the old program names. Once all the disk names have been read in and stored, however, things will pick up because the new programs will load freely at the end of the current index.

It is possible, by loading a large index to another large index, to exceed the capacity of your memory. If this occurs, you will be told so. At this point your index is incomplete and should not be used. Since the disk names are loaded first and the program names last, you will have incomplete records of some disk directories. You should note, however, that the upper memory limit of INDEX is variable, and may be adjusted under PROGRAM VARIABLES. If you have more memory than this value is set to, you can make it larger to get more room, though the default value for this memory size is all that you have.

PROGRAM VARIABLES

This is a special mode that has it's own menu. It allows you to program some of the values that are used by INDEX. Along the left side of the screen will be the functions you can change, and along the right side are the current values being used. To escape from this mode, just hit <BREAK>. Each item will be covered separately.

SET TOP-OF-FORM

On the right hand side of the screen you will see a number listed as the number of LINES LEFT. This means the number of lines left to be printed by the line printer on what INDEX considers to be the current page. INDEX can't know where the actual end of your page is, but it assumes your paper has 66 lines per page and that it will print on 56 of them. If the number of LINES LEFT is less than 56, INDEX is not at the top of a page. If you type S to set the top-of-form, INDEX will call the current position of the paper top-of-form and set the number of LINES LEFT to 56. All that SET TOP-OF-FORM does is set the number of LINES LEFT to 56.

### FORM FEED

This command will cause the paper in the line printer to advance to the top of the next page. If the LINES LEFT counter shows the number 56, this will be a whole page of 66 lines. If the number is less than 56, INDEX will skip the number of lines shown plus 10, which is the size of the bottom margin.

### LINE FEED

This command will cause the paper in the line printer to advance one line. It will also cause the LINES LEFT counter to decrease by a line at a time until it reaches zero. This command is used to position the paper in the printer.

### PRINTER ON/OFF

This is the way to enable line printing elsewhere in the program. In both the LIST and FIND modes you will see a printer message that tells you if the printer is on or off. When the printer is defined as off, no characters will be sent to it. If the printer is defined as on, what appears on the video screen in the LIST and FIND modes will also be sent to the line printer. On the right side of the screen in the PROGRAM VARIABLES mode will be a message telling whether the printer is currently defined as ON or OFF. If you hit the P key the current state of the printer will change. If it is off when you hit P, it will change to on. If it is on when you hit P, it will change to off. Regardless of whether or not your line printer is actually plugged in and turned on, the PRINTER ON/OFF selection is what determines if INDEX will even try to use your printer. If you tell INDEX to turn the printer on and you don't actually turn your printer on, INDEX will lock up when it tries to print something.

INDEX uses the standard device control block for printer I/O. Using these control blocks makes it possible to define a custom printer driver for people with non-standard printers. If you have such a print driver, load it from DOS READY before entering INDEX. INDEX will also allow you to protect memory for this driver (see CHANGE MEMORY LIMIT). Using the device control block can cause problems too, however. INDEX does it's own formatting for page lengths. If you are using a printer routine that also wants to format your printer output, they can work against one another. If your custom printer routine wants to paginate, don't let it.

### NEW PAGE WAIT

On the right side of the screen is a line that says WAIT: and either YES or NO. Let's assume it says NO. This means that when INDEX gets to the bottom of a full page of printing, it will go to the top of the next page and keep right on going without waiting. This is fine if you are using continuous paper. But if you are using single sheets it's a little bit awkward. If you want INDEX to stop after each page and wait for you put in a new sheet, you can tell it to. To turn the wait mode on, just hit the W key. The WAIT message on the right side of the screen will then say YES. To turn it off, again, hit W again. It toggles like the PRINTER ON/OFF control. With the wait mode enabled, INDEX will finish printing a page, skip to the theoretical top of the next page, and then stop. The screen will tell you to put in a new sheet of paper and hit <ENTER>. INDEX will then continue printing on the new sheet.

### CHANGE LINE FEED CODE

Because all printers don't accept a real form feed character, INDEX does it's form feeds by sending a series of line feeds to the printer to get it to go to the top of the next page. But not all printers use the same line feed code either. On the right side of the screen you will see the decimal value of the current LINE FEED CODE. An ASCII line feed is 10. An ASCII carriage return is 13. Some printers expect 138 or 141 for line feeds. INDEX will let you use anything you want. Type C to change the line feed code. You will then be asked for the decimal value you want to use. You may specify any number between 0 and 255. After hitting <ENTER>, this number will be displayed on the right side of the screen, and will be used from then on as the line feed character.



### INVISIBLE PROGRAMS

On the right side of the screen you will see a message that says INVISIBLE: and either YES or NO. If it says NO, then INDEX will not find any programs that are defined as invisible when you BUILD AN INDEX or DISPLAY A DIRECTORY. If the message on your screen says INVISIBLE: YES, then the invisible programs will be read and stored in your index with the visible ones. To change this message, hit the I key. This will cause the invisible status to change. If it was YES, it will become NO. If it was NO, it will become YES. If you are not familiar with invisible programs, look up invisible status in your DOS manual for more information.

### CHANGE MEMORY LIMIT

On the right side of the screen you will see a number represented as the TOP OF MEMORY. This is the highest memory address INDEX will use to store programs. If you have some special keyboard driver or printer driver you want to run in upper memory, you can set a memory limit for INDEX to protect this. When INDEX first comes up, this value is set to the full amount of memory you have in your machine. To change this number, hit M. You will then be asked for the decimal value you want to use. You may enter any number you like, but numbers less than a minimum (which is variable and lower than you will ever want to use) will be changed to that minimum so as not to crash the program. Values higher than 65535 exceed the memory capacity of the TRS-80, are not allowed, and for technical reasons will be interpreted as the minimum.

### ERASE CURRENT INDEX

If you should find it necessary to eliminate every program and disk in the index, you can do that quickly with this command. It takes two hands (or at least two fingers) to type an asterisk, so you aren't likely to do it by accident. When you type an asterisk, you will be returned to the command screen with no disks and no programs in the index.

### DISPLAY A DIRECTORY

This function is selected from the main command screen. To see the directory on the disk in any drive, type the number of that drive. In this mode the program names are not stored in the index, they are just displayed on the video screen. This function uses the same search sequence as the BUILD AN INDEX function to identify any type of operating system. The number of free grants and the type of disk will also be displayed.

### EXIT THE PROGRAM

This is the way back to DOS READY. After typing X you will be given a warning message. If you have made any changes in your index in the current session, be sure to have saved it on disk. It is also necessary to have a system disk in drive 0 before you return to DOS READY. If you decide you don't want to leave INDEX yet, hit any key except V and you will return to the command screen with your index intact. If you are ready to leave INDEX, hit the V key to verify your exit request and you will be returned to DOS READY.

