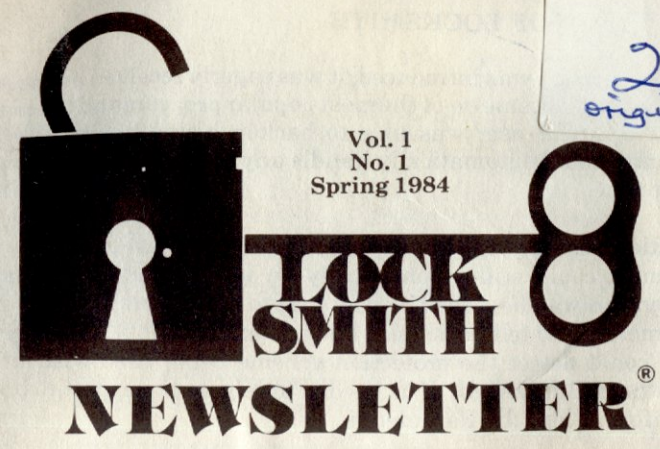


2
original

Vol. 1
No. 1
Spring 1984



© 1984 Omega MicroWare, Inc.

IN THIS ISSUE	
Publishers Note	1
Evolution of Locksmith	2
Locksmith 5.0	2
Locksmith 6.0?	3
Why ParmS	3
Helpful Hints	4
Locksmith Compatible Hardware	5
The Omega Policy	6
Loading LPL Files	6
Using Quickscan	8
What Do I Do When....?	11
Letters to the Editor	14
Future Articles	17
Current Locksmith 5.0	
Revision Level	17
Track Syschronization	19
Users Manual Errata	21
Disk Backup Seminar	21
Renewal Form	22
ParmS Listings	23

Publishers Note

WHY THIS NEWSLETTER

With the release of Locksmith in 1980, the cycle of copy protection schemes and the ability to back-up your programs, began. For the user with no technical expertise, the inability to back up VisiCalc or DB Master represented a real problem. Some manufacturers provided back ups at exorbitant prices while others provided NO backup at all.

For the expert computerist, the primitive techniques provided little challenge. They simply went into memory and removed the protection, making a copyable unprotected program.

As computer users got smarter and began to learn to defeat simple copy methods, software publishers got tougher with their protection methods.

In order to meet the need of the ordinary user (those of us who lack technical knowledge of the disk operating system), Locksmith came into being. This newsletter is an outgrowth of Locksmith, its popularity and its growth.

Omega MicroWare Inc.
177 W. Hintz Rd.
Wheeling, IL 60090

BULK RATE
U.S. POSTAGE
PAID
Permit No. 127
WHEELING, IL 60090

Scanned by cvxmelody

<http://www.cvxmelody.net/AppleUsersGroupSydneyAppleIIDiskCollection.htm>

EVOLUTION OF LOCKSMITH

When Locksmith was introduced, it was eagerly received by the Apple user and soon became one of the most popular programs on the market. Now, the novice user was able to backup his software. The early Locksmith could automatically handle any protection scheme known at that time.

Protection schemes continued to get more complex. In many cases, Locksmith could still automatically do its job, but in others user intervention was needed. The user had to enter certain codes, called "Parameters", to tell Locksmith how to handle its job. As always, the expert could detect the protection scheme being used without help from others. But the rest of us needed to rely on the expert to provide that information; the Parameters.

The Locksmith folks tried many ways to disseminate Parameters. Providing them over the telephone by setting them up on the Source or providing a user accessible bulletin board, sending Parameters via postage paid envelopes, and others. Finally, the magnitude grew until the service was not only insufficient, but did not meet the Locksmith owners needs.

At the same time, protection schemes themselves outgrew the ability of the then current release of Locksmith. Although it had been updated several times, Locksmith was finding it more difficult in copying the newer protection schemes.

LOCKSMITH 5.0

In late 1982, the developers of Locksmith began to rethink the theory and methods of protection schemes, Past, Present, and FUTURE.

They wanted Locksmith to provide the uninformed (and uninterested) Apple user a realistic way to back up his programs. In addition, they wanted to give the expert a tool with which to analyze, evaluate, and understand how protection schemes were evolving.

Locksmith 5.0 does both!

If you use the manual and have a good knowledge of the Disk Operating System, you can develop the needed parameters all by yourself.

Omega realizes that many of you don't want to mess around with a manual, ours or anyone's else's nor do you want to take the time to figure out parameters.

So, we elected to use this newsletter approach to provide Parameters, answer questions, teach and act as a general forum for Locksmith owners to air their views, their gripes, and provide us with a way to answer your needs.

LOCKSMITH 6.0?

Will there be a Locksmith 6.0?

Frankly, we don't think so as we believe Locksmith 5.0 is versatile enough to handle anything new that comes up.

But, protection schemes are unknowns. If there is a need, we will (at a fair price) provide Locksmith owners the means and ability to upgrade to any future version.

In fact, this newsletter contains patches that allow you to upgrade your Locksmith 5.0 to the current release level at no charge.

WHY PARMS?

We have received many questions asking why parameters are still necessary to back up some programs. Let us explain a few things about copy protection.

First, it is necessary to explain a few terms that we will be using. A copy method refers to the way in which a disk is copied. This includes all parameters and algorithms which are used, in addition to specifying which tracks will be copied and whether or not they are synchronized or nibble counted. An entire copy method is contained in each LPL file which we supply to back up a specific disk.

An *algorithm* specifies a method (a sequence of steps) of analyzing a disk once it is read. After reading a track on a disk, it is analyzed by the various algorithms. While writing 5.0, algorithms were included for just about anything we could think of. At this point, there are algorithms which have never been used, but were added to try to allow for all contingencies. Analysis of a track is usually performed by several different algorithms, executed in a specific sequence. While there is a default algorithm stack, sometimes it is necessary to alter the algorithms used to back up a specific disk.

The term *parameter* has been used to mean several different things over the last few years. When we use the term, it is in reference to a specific location of the program which may be changed to allow the user to customize some of the different algorithms or subroutines in Locksmith. Most parameters are used to modify the various algorithms, although some may be used to change other functions, such as read, write, etc.

A copy process refers to the way in which the disk is copied. As a track is copied, it may be copied normally, synchronized, or nibble counted. On any one disk, any combination of these processes may be used on any number of tracks.

When Locksmith was first introduced several years ago, the copy protection techniques were not very sophisticated. Over the last few years, the techniques have become more and more sophisticated. The state of the art has progressed (digressed?) to the point where it is no longer possible to back up all disks using the same methods. There are so many combinations of methods that may be used, that it is virtually impossible to predict just how a disk may be copied. Some disks use nibble counting, some use synchronization, etc. Using these two methods alone, it would be very time consuming to synchronize and nibble count every track on a disk. For this reason, it is necessary to determine which tracks contain valid data, and what the proper means are to copy them. One disk may use half tracks, for instance, which must be copied differently than normal tracks. Some disks look for specific sequences of nibbles on a disk, requiring special analysis to insure that they are transferred intact, during a copy operation.

Locksmith 5.0 is the most sophisticated nibble copy program on the market. It requires fewer parm changes and fewer algorithm changes than any other program. Copy protection has advanced to the point where it is no longer possible to back up all disks without some changes. Because of this, we have introduced LPL files, to make it easier to perform these changes. In the parameter section of this newsletter, it may be seen that an LPL file contains all of the information necessary to back up a disk. Once a file is loaded, it will execute and copy an entire disk, making all parm, algorithm and process changes as necessary for any and all tracks. This is in contrast to other programs, which require entering all changes by hand, as well as making multiple passes on a copy, in order to insure copying all tracks. From a user standpoint, it makes a very difficult process, simple and easy.

We have reached a stage where we will not likely again see copy programs that require no parm, algorithm or process changes. The protection methods are simply too varied. This being the case, we will endeavor to make it as easy as possible, from a user standpoint, to make these changes and back up the disks.

Included with this newsletter is a renewal form so you may continue to receive future issues. You may also order a disk with the parms already keyed in so you don't have to spend the time to do so yourself.

HELPFUL HINTS

Some suggestions that we've found useful:

* When copying a disk which is recorded with a spiral track technique, it is recommended that the target disk be either a new (unused) diskette, or that it be previously erased.

* If, when copying from drive 1 to drive 2, the copy repeatedly fails with errors of code 5 or code 8, try reversing the order of original/copy drives.

* If in doubt about which of your drives to use for original and which to use for copy diskette, use the slower drive for the copy (target) disk. This will reduce the number of track-overwrite corrections, and make the copy operation faster (and more accurate).

* When working with an unknown disk for the first time, it is most useful to examine it with the quickscan function first. Then examine any interesting tracks with the nibble-editor.

* When developing LPL parameters of your own, it is best to use one of the parameters supplied on the Locksmith disk or in the newsletter, and modify it to suit your needs. Also, the nibble-editor is very useful for testing the results of individual algorithms on a track to best determine which ones will be of use on a particular track.

* Write-protect your original diskette (and your Locksmith diskette) to prevent accidental erasure while attempting to copy it. This is especial true for those of you using Locksmith with only one drive.

* Now here's a useful one: If you accidentally clobber your Locksmith diskette using the disk speed utility, don't despair -- only track zero has been destroyed. Use your backup copy of Locksmith 5.0 to copy track zero from the backup disk to the original disk (using the 'B' function of the main menu). You will have fixed track zero which was erased by the disk speed utility.

* If a disk copies and shows many 8 errors, it may be necessary to change the number 1 parm (DATAFB) from a 2 to a 1. After making this change, try to copy the disk again.

LOCKSMITH COMPATIBLE HARDWARE

This is a list of known compatible hardware.

Locksmith compatible RAM boards:

Ramex 128K - Omega Microware
Saturn 128K, 64K, 32K - Saturn Systems
MPC 32K - MPC Peripherals
all 16K RAM boards

Locksmith compatible //e auxiliary memory boards:

Apple //e 64K 80-column/aux. mem board

Locksmith compatible disk drives:

Apple
Mitac
Amdek 3-inch micro-floppy drive

Locksmith compatible disk controllers:

Apple

THE OMEGA POLICY

The copyright law allows the creation of archival copies of computer software which is owned by the Locksmith owner. Locksmith is sold with the understanding that the purchaser will not use the program to generate disks of copyrighted programs for sale or distribution. Should the program be misused, Omega MicroWare, Inc. will assist in the prosecution of violators at the copyright holder's request.

LOADING LPL FILES

In the back of this newsletter, there is a list of parameters that may be used to back up different programs. Due to the number of these parms, it is best to store the parameters on a diskette. On the Locksmith diskette which you have received, there are a couple of different parameter files. Each of these files represents the parameters required to back up 1 disk. The parameters which we are supplying in this newsletter are actually entered in just a couple of very large files, to save disk space. Loading parameters from this type of file is slightly different from loading an entire file.

First, we will look at a typical parameter entry for a program in the file. For this example, we will use the default Locksmith LPL file. The file as currently shipped looks like this:

```
**DEFAULT LOCKSMITH LPL FILE  
READ  
CHANGE SS DHDR  
TSTART DOS PAT1 STAT 1  
IF FAIL 1 THEN TSTART LONG SS  
TEND REPEAT  
DISPLAY  
TSTART BSPACE  
TEND FSPACE 3  
VSTART NORM  
LABEL 1:WRITE STAT 1  
IF SUCC 1 THEN GOTO EE  
SHORTEN ALL EQUAL BY 2 STAT 5
```

```
IF FAIL 5 THEN GOTO DD  
CLEAR ERROR : GOTO 1  
LABEL DD  
SHORTEN ALL CENTER BY 2 STAT 6  
IF FAIL 6 GOTO DA  
CLEAR ERROR : GOTO 1  
LABEL DA: *SHORTENING FAILED  
LABEL EE: *WRITE SUCCEEDED  
***
```

There are two very important things to note in this file. At the very beginning of the parameter listing, the first line contains 2 asterisks, followed by the name of the parameter itself. At the very end of the listing, there are 3 asterisks. These asterisks serve a very important function. They are the delimiters that show the beginning and end of a section of parms. By using them, it is possible to load a portion of an LPL file rather than the entire file. This allows us to string a large number of parameters into one file.

When loading a portion of an LPL file, the following procedure is used. First, enter the Text Editor of Locksmith. This is done by pressing "T" from the main menu. You may or may not be asked to insert your Locksmith disk and press the space bar, depending on whether or not the Text Editor is currently in memory. Once the Text Editor is loaded, you will see the Editor menu. To load parameters, press "L". You will then be prompted to enter the file name. You should then enter the name of the disk file that contains the parameters, and press RETURN. Next you will be prompted for the Search String. This is where the asterisks become very important.

The Search String works in the following manner. In this example, we would search for the search string "DEFAULT LOCKSMITH LPL FILE". Locksmith would then search through the entire file looking for 2 asterisks followed by the search string we have specified. It is very important that there be no spaces between the asterisks and the string we have specified, and that we not tell it to search for the asterisks. It will automatically search for the 2 asterisks followed by our search string. After finding the search string (which is the beginning of the parameters we are interested in), it will start loading the parameters into the buffer from that point. It will continue loading parameters until it finds 3 asterisks in a row. This signifies that we have reached the end of the parameter listing. At this point, control is passed back to the editor. From this point, we may then press "B" to backup the disk or use any of the editors other features.

This feature is very useful, but must be performed correctly. Locksmith looks for 2 asterisks followed by the search string, and will then start loading and continue loading parameters until 3 asterisks are encountered. When making up a parameter file, we suggest that you always use this format for saving parameters. This way, large parameter files may be used.

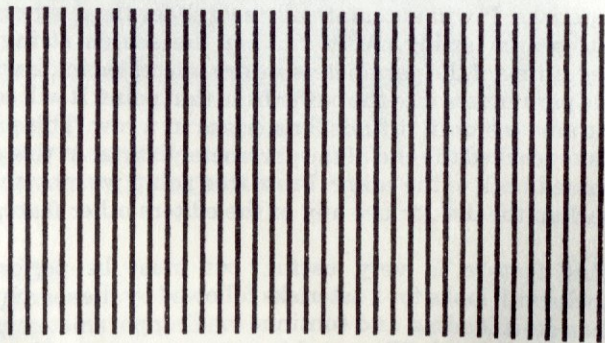
USING QUICKSCAN

Quickscan is not only one of the most useful features of Locksmith 5.0, it is also one of the least understood. In this article, we are going to try to give the user a better feel for just what this function does.

Quickscan is a fast way to determine what type of data is stored on a diskette. To use this feature to its fullest potential requires some practice, so we are going to be looking at a few examples of typical disks to help describe what is happening. First, it is necessary to explain what the function does. Very simply, Quickscan will scan a disk from a starting track to an end track, displaying areas of valid and invalid data. After selecting Quickscan, you will be prompted for the correct disk drive, starting track, ending track and increment. After these are entered, the drive will turn on and very rapidly seek (move) to the different tracks displaying the graphic representation of the data.

Perhaps one of the least understood aspects of this function is the screen display itself. Across the bottom of the screen, are the numbers signifying the tracks from \$00 to \$23. Above each track number, a display of track data will be drawn. When scanning a track, the track data is read into the nibble buffer. After it is read, it is analyzed to determine content. This is done by looking at the first 41 nibbles of data and determining if there are any invalid or self-sync nibbles in the range. If either invalid or self-sync nibbles are found in this range, a dot will be drawn at the bottom of the screen just above the track number we are working on. If no invalid or self-sync nibbles are found, no dot will be drawn. We then analyze the next group of 41 nibbles and draw a dot if necessary above the previous dot. We continue analyzing and drawing until we have completed slightly more than 1 revolution of the diskette. This will give us a vertical line from the bottom of the screen to the top, showing wherever invalid or self-sync nibbles are located. With a little practice, we can inspect this line and determine how the track is written with data.

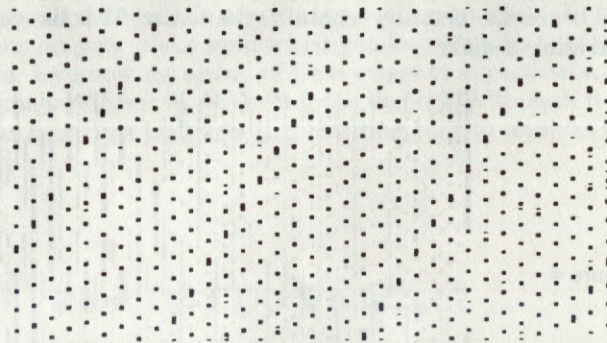
Figure 1



```
TRACK  HEX 000C0000000000000111111111111112222
NUMBER HEX 0123456789ABCDEF0123456789ABCDEF0123
```

For our first example (figure 1), we will look at a blank or non-initialized diskette. By referring to the diagram, it can be seen that all tracks are composed of solid lines. This is because there is no valid data written, and the disk contains nothing but invalid nibbles. Conversely, if the entire track were composed of valid data, no line would be seen at all, since no invalid or self-sync nibbles would be found. For the same reason, doing a Quickscan with the disk drive door left open will also display a solid line, representing no valid data.

Figure 2



```
TRACK  HEX 000C0000000000000111111111111112222
NUMBER HEX 0123456789ABCDEF0123456789ABCDEF0123
```

Our next diagram (figure 2) shows a normal 16 sector diskette. After inspecting a track, it can be seen that there are areas of valid data (sections of no dots) as well as areas of self-sync data (sections of dots). These areas are repeated in a very steady pattern around the track. If you look closely, you can determine where each sector starts and ends. The longest section of self-sync also represents the start of the track, since there are more self-sync at the start of the track than between any other sectors. Since we read slightly more than 1 full track at a time, sometimes we will actually count 17 sectors instead of 16. This is due to the overlap.

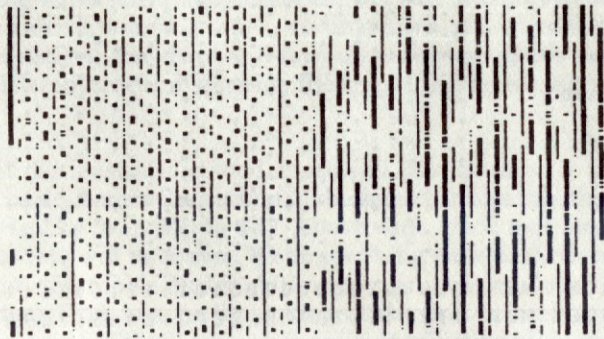
Figure 3



```
TRACK  HEX 000C0000000000000111111111111112222
NUMBER HEX 0123456789ABCDEF0123456789ABCDEF0123
```

We also have a diagram of the same 16 sector disk (figure 3), but instead of reading the normal tracks, we have read from track .5 to 21.5. This displays the regions of the disk between the tracks that were written. It is still possible to see some of the sectors from the adjacent tracks, but the pattern is not regular like it is in the other diagram. In some cases, the areas of self-sync or invalid data actually run together, showing a long line. These 2 diagrams were shown this way for clarity, but they could be combined. If you scanned from track \$00 to track \$22, incrementing by .5, we would see both tracks and half tracks. By looking at the display, we should be able to determine which tracks actually contain the data. This is very useful in analyzing a disk.

Figure 4



```
TRACK  HEX  000C0000000000000111111111111112222
NUMBER HEX  0123456789ABCDEF0123456789ABCDEF0123
```

The final example (figure 4) will show how this may be done. The disk was read from track \$00 to track \$22, incrementing by .5. This shows all tracks and half tracks. First, we can look at track 0. Unlike the previous example, this is not a normal 16 sector track. Instead, there are a few regions of valid data, but most of the track is invalid or self-sync (the long line). If we start examining the data past this point, we can see that the pattern is very regular on the lines which represent full tracks, and irregular on the half track lines, up through track \$11. This indicates that there is valid data on tracks \$01 through \$11. It should also be noted that these appear to be normal 16 sector tracks.

After track \$11, the data starts to look different. Starting at track \$12 and continuing through track \$22, we can see regions of valid data (no lines drawn) on all half tracks as well as full tracks. If we look closely, we can also see a pattern to these areas of valid data. If you will look at track \$12, you can see a couple of long areas of valid data. If we then look at track \$12.5, we can see that the areas of valid data on this track appear to follow the valid data on track \$12. Since the data is represented from the bottom to the top, we can see that as an area of data ends on track \$12, an area of data starts on track \$12.5. This pattern continues through track \$22. This type of pattern represents a spiral track. The data on track \$12 is not adjacent to data on track \$12.5, but is offset. This is the entire purpose behind spiral tracking.

By scanning this disk, we have seen that there is data on tracks \$00 through track \$11, incrementing by 1, and on tracks \$12 through track \$22, incrementing by .5. We now know which tracks will need to be copied, as well as knowing a little bit about the format of the data on the disk. With a little practice, we can learn to spot some common patterns on disks. For example, it is very easy to recognize 16 sector disks, as well as spiral tracks.

This feature of Locksmith 5.0 is one of the easiest methods to get "a feel" for what type of protection is on a disk. Analyzing the display will become easier with a little practice, and we recommend that you do Quickscans on your disks even when you know which tracks are used for data. By knowing ahead of time where to expect data, it is easy to become proficient in analyzing the display. Although it will take practice, it is time well spent in learning how to analyze disk protection.

What Do I Do When.....?

Many people have asked 'What do I do when the program I want to copy is not on the parameter list'. My first question is how familiar are you with the way information is stored on a disk. Can you tell me what "D5 AA 96" is?

If you can't answer this question then you aren't familiar enough with the way information is stored on a disk. Don't despair however, it is really not that hard to learn. We will give you a few basics now and recommend a good book to read that will also help you learn what you need to know.

The first thing we need to do is to be able to recognize specific information on a normal diskette. This will help us to determine what the differences are between the normal disk and a copy protected one. Since the fastest way to learn is by doing, we will now proceed to do.

Take a copy of your DOS system master disk that came with your disk drives. Put this disk in drive 1 and turn on the computer. After the drive light goes out and the basic prompt appears, remove this disk. Now insert a blank disk into the drive. Type FP and press the return key. This tells the DOS to throw away the current BASIC program. Now type INIT HELLO and press the return key.

What we have done is this: We have created a test disk for us to look at that has nothing on it except for the DOS on tracks 0,1 and 2. It also has the CATALOG and the VTOC on track 11. Don't worry if you don't know what this means. It is not necessary that you understand how the DOS uses the specific file information on a disk in order to copy it.

Next remove this disk from the drive and insert your Locksmith disk. Turn the power off, then on again to boot the disk. If you know how to boot a disk without turning the power off you can also do that. Follow the prompts until you get to the Locksmith main menu. Press N for Nibble Editor. Follow the prompt. We will take a look at what the Nibble Editor is used for.

Now remove your Locksmith disk from the drive and insert the test disk we created earlier. You should be in the Nibble Editor with what looks like a lot of junk on the screen. Press and hold the key marked CTRL and press the key marked R. Locksmith will respond just below the status display with the word TRACK:. Press the number 3 followed by the return key. Now Locksmith, on the same line, will ask DRIVE:. Press the number 1. We have now told Locksmith to read track 3 from the disk in drive 1 into the Nibble Editor. A few seconds after the light on the drive goes out you should see some new information in the nibble buffer.

We will now discuss what all this stuff in the buffer is and what it means to you. If you are not familiar with how to move around in the Nibble Editor, you should read the section in the manual that discusses this. It starts on page 41 of the manual. In the buffer you should see a lot of numbers and letters. Some of these should be in a white box (inverse). If you don't see any of these, move down into the buffer until you do. Since we created this disk together, I know what you are going to see and can, hopefully, give you a simple explanation of what we are looking at.

The letters in the white boxes are synchronization bytes or sync bytes, for short. Sync bytes are used by the disk controller card hardware to make sure that the data it is reading is valid. Data is actually stored on the disk as a series of 1's and 0's. A one means the bit is turned on and a zero means it is turned off. Since the hardware just starts reading the ones and zero's when it comes up to operational speed it more than likely will be in the middle of a byte of information. A normal byte is eight bits of information. A self sync byte is a 10 bit byte. It has the normal eight bits of information followed by two zero's. Since this is a disk created with a non copy protected DOS the sync bytes will be FF. You should see FF's inside the little white boxes. Sometimes it will be something different than an FF due to the drive writing in the middle of another byte or writing before it is fully up to speed.

Following the self sync FF's you should see 'D5 AA 96'. This is the address header for a normal DOS disk. We must learn to recognize this pattern easily. This is the pattern DOS looks for when it wants to read an address field. Following the address field header we have several other bytes of information. The next two bytes should be 'FF FE'. This is the disk volume number in a special encoded form. I will give you the name of a book later that goes into more detail on this information. This particular sequence 'FF FE' stands for value 254. I know this because we just took the default volume number when the disk was

created rather than specifying one. Also there is a nibble decode table in your Locksmith manual on page 134 and 135. There is also a track decode table in the same place. The next two bytes should be 'AB AB', this stands for track three. The next two bytes stand for the sector number, the next two bytes are the checksum for the previous bytes in the address field. The last three bytes are 'DE AA EB'. These three bytes are the address field epilogue or trailer. The third byte 'EB' is never checked by DOS and sometimes gets overwritten when the sector is rewritten to the disk when a file is saved.

Following the address field should be some more sync bytes (in inverse) and, then, the data field header or prologue. This is 'D5 AA AD'. All versions of DOS use the same address field header. Following the address field header is the actual address field data. Since this sector on the disk is empty, all sectors are empty except for the tracks mentioned earlier. All the data bytes will be '96'. This is the way a '0' is encoded in the current DOS version. Following the data field is the data field epilogue or trailer. It is the same as the address field trailer, normally, 'DE AA EB'. In this case the 'EB' is always checked. If it is missing, DOS won't read the sector.

All of the above information is for a normal 16 sector disk since that is what we created to look at. If the disk had been 13 sector then the address field header would have been 'D5 AA B5'. All the other information would be the same except for the data fields. They will contain 'FF' in an unused sector.

Now that you know what to look for, get out your Locksmith manual and go to the section on the Nibble Editor and try out all of the commands. There are commands to automatically find a specific string for you. There is also a command to decode the address fields for you.

This information, address field and data field headers and trailers are normally the first things that are changed on copy protected disks. They may also change the number of sectors on a track, but Locksmith doesn't care about that. The sync bytes are usually changed on protected disks. They still will show up inside the little white boxes, however. After you have become familiar with the Nibble Editor on this one track, look at data on track 0. See if you can find the same fields on this track. If you can, then it is time to try a few copy protected disks.

If you examine the furnished parameters, you will see that they all set patterns for address fields and, occasionally, for sync bytes. Some sync bytes read as sync more reliably than others.

This should be enough information to you get started. There will be more articles on this subject in future issues of the newsletter. Beneath Apple DOS is a book that explains the DOS in much greater detail and is published by Quality Software.

Letters to the Editor

(Well, not really letters to the editors. Every time we see a new computer magazine that has a letters column, we wonder how those people knew to write to a magazine they did not know existed. These are all questions either asked of our customer service department or mailed to us. In future issues, we will answer letters actually sent to the newsletter.)

Dear Sirs,

Locksmith 5.0 will not backup my DB Master IV. Your advertising leads me to believe that it will back up anything.

H.F.
San Jose, Ca.

Dear H.,

If you do a Quick Scan of your DB Master, you will see that tracks 0-5 are normal when the scan is set at increments of 1 starting at track 0. You will also see that tracks 6-22 look like straight vertical lines, much different than tracks 0 through 5.

Doing a Quick Scan of half track increments will show you a typical synchronized pattern beginning at track 6.5 through 21.5.

So, set the copy mode to track 0 - 5 at normal and copy those tracks. Then, set the copy mode to synchronized and copy tracks 6.5 to 21.5. This will give you a back up of your program.

Dear Sirs,

I cannot copy my friend's "Wizardry" even though he bought it over a year ago. Shouldn't my new Locksmith 5.0 be able to copy it?

L.L.
Long Beach, Ca.

Dear L.L.

It is illegal and unethical for you to copy your friend's "Wizardry" or any other program which doesn't belong to you. It is also illegal to even use your friend's backup copy even though he has a legal right to make one.

Don't be a thief!

Programmers work long and hard making, marketing and maintaining commercial programs. They deserve to be paid for their labor.

Public law 96-517 provides that it is not a violation to make a copy for archival purposes. However, it does not allow you to give away those copies or your original while you keep a copy. The law is simple and straightforward. Copies you make are for archival purposes only.

If a program is worth using.....BUY IT, don't steal it.

Dear Sir,

I am thinking of buying one of those peripheral cards that takes a picture of the program in memory. Are they worth it?

B.U.
Bethesda, Md.

Dear B.,

Cards which claim to copy programs from memory actually only copy that part of the program that resides in memory at the time the picture is taken.

Unless the entire program resides in memory after loading, each section must be saved from memory after it has been loaded. Even then, you will have to make modifications so your copy can recognize each module when the previous program module requires it.

While this works for some small programs, any program that accesses the disk will operate better using Locksmith instead of a copy card.

Dear Sirs,

When should I try track synchronization? Nibble counting? How can I detect when spiral tracking is used?

R.W.
Ogdensburg, NY

Dear R.,

We get a lot of questions like these, mostly from the less technically proficient Locksmith user. To determine beyond a shadow of a doubt what type of protection technique is being used on a particular disk, it is sometimes necessary to decipher the code on the protected disk.

Valuable "clues" can sometimes be acquired by other means such as listening to the disk boot. Does the head step across the tracks in a smooth manner, implying track synchronization? Does the head stop on a track for extended analysis during boot and then continue? This may indicate nibble counting on the track. What track is the head on when the boot fails on the copy disk? In a future newsletter issue we will discuss methods to determine the current track while the disk is being used.

The quickscan function can also provide many clues. Are half-tracks used? Does the data on adjacent tracks seem to align? Are any tracks erased or empty? Do any tracks appear to be in a different format than other tracks? If the quickscan display indicates interesting tracks, what does a more detailed look with the nibble editor show for these tracks? Does the track read reliably? Is it abnormally long or short? Finally, try copying the disk. Many of the Locksmith parms provided in this newsletter were determined after many hours of trial and error.

In future issues we will show you how to determine protection techniques and back up some of the most difficult disks available, step by step, from the time we open the package of the original to the time we successfully boot the back up disk.

If all this detective work is not for you -- if you are not interested in how protection works but only in how to backup your disks then the newsletter department on Locksmith parameters is for you. Each issue, "how to copy" parameters for the most often requested protected software are presented. This charter issue contains a good selection of parameters. More will follow in future issues. These parameters are provided by the Locksmith author and Locksmith support team, as well as from our Locksmith users.

The Locksmith Newsletter solicits parameter information from it's readers. It is impossible for us at the Newsletter to determine parameters for all protected software. We need your help. In addition to "How to copy" parameters, any suggestions or helpful hints which you think other readers might find useful would be appreciated.

Dear Sirs,

"I don't want to learn any thing about software protection. I just want to back up my disks using Locksmith."

S.G.
Highland Park, Il.

Dear S.,

Locksmith is a tool, much like a pipe wrench is to a plumber. As you know, the pipe wrench won't fix your leaky faucet by itself. In this Newsletter, we hope to show you how to be your own "plumber", using Locksmith effectively for all your disk copying needs.

FUTURE ARTICLES

Future articles planned for the newsletter are:

"Inside LPL: effective techniques"

"Weak bits and unreliable data"

"Attacking the spiral track"

"Delatching the shift register"

"The power of the Nibble-editor"

"Nibble counting: how it works"

"The role of self-sync in protection"

CURRENT LOCKSMITH 5.0 REVISION LEVEL

This regular department of the newsletter contains the current revision level and the patch to upgrade your Locksmith to that revision level.

The current Locksmith 5.0 revision level is "F". The revision level of your Locksmith is displayed at boot. If your Locksmith is not at the current level, you can upgrade it to the current level yourself, using the Locksmith disk patcher function. DO NOT send in your Locksmith disk for revision level changes.

The following data must be entered into an Apple DOS text file. You can enter it with the Locksmith text editor or any other text editor which creates text file output.

On your Locksmith disk is a file called "PATCH". You can add to this file to bring it current and then apply it to your Locksmith diskette. Your Locksmith 5.0 will then be up-to-date and at the current revision level.

Select the text editor from the Locksmith main menu by pressing "T". When in the text editor menu, enter "C" to display the catalog. Load the PATCH file by pressing "L". Press the return key when prompted for "search string". After the file is loaded from disk, press "E" to enter the editor. Add, to the end of the file, any lines which are missing from the listing below.

When finished entering data, press "ESC" to get back to the text editor menu and save the file to disk with the "S" key (save file). Then press the "P" key to apply the patch. If any mistakes were made keying in any information, the line containing the error will be automatically displayed, allowing you to make changes to it. If any corrections are made, be sure to save the file to disk again. If no errors are detected, you will be prompted to insert your Locksmith diskette and press the spacebar. After doing so, your Locksmith disk will be updated to the current revision level. Reboot your Locksmith disk and verify the new revision level which is displayed at boot. If the patch is successfully applied, you may want to apply it to your backup disk as well.

It is recommended that the user maintain his Locksmith disk at the current revision level in order to benefit from the most recent enhancements.

The current patch follows immediately:

- * THE FOLLOWING LOCKSMITH 5.0 PATCH
- * WILL PATCH LS 5.0 REV A,B,C,D, OR E
- * DISKS TO REVISION LEVEL F. NOTE THAT
- * THIS MUST BE RUN WITH A LOCKSMITH OF
- * REVISION LEVEL C OR LATER, BUT THE
- * PATCH CAN BE APPLIED TO ANY LS 5.0.

- * IF ERROR ON NEXT LINE, SEE NOTE ABOVE

B8100B49EA

FDF02DCDC

- * IF ERROR ON NEXT LINE, CHECK W/PROT.

3D1001EEC2

97100C58D3

- * NOW AT REVISION LEVEL B

E50C0973DA7555B65134E65C

9E0C097BF1345DD980E82BE2

A10C0983AF3CA09682218BAA

7A0C098B4BAAC17EAA

DA0C09A37C

2B0C09E409C3

650C09EA8A

720C082A510D

740C0A4C3E

- * NOW AT EARLY REV C

740C0A5321

740C0A582A

740C0A6517

740C0A6A18

740C0A7301

CA0F064E8D

- * NOW AT REVISION LEVEL C

4B080EEB5B60ED5A62F4317EF3
AA080EF4447BF64375FA3A17FC4D2EEF
88080D0101B6D30F8804C7C409BADB
52080D0C0FE10EBB
CD0F064E8A

*

220F06C30DE5

2B0F06CE67AF1DBECB23D118A6

C80F06D7CE08DA57195C915DD4

- * NOW AT REVISION LEVEL D

81090864E4

CC0F064E8B

- * NOW AT REVISION LEVEL E

87040B67EF

E5040B6C86

03040A1C11

CF0F064E88

- * NOW AT REVISION LEVEL F

TRACK SYNCHRONIZATION

Normally, when tracks are written to a disk, they are written independent of each other. Apple disks are soft-sectored, so there is no specific placement of sectors around a disk. When either reading or writing to a track, the read/write head on the disk drive is stepped to the correct track. The disk is rotating, and when the correct data is under the read/write head, the data will be either read or written. The position of the data on the track is not critical. If the sectors on a disk were moved around in relation to each other, it would make no difference.

Several years ago, a new form of copy protection was introduced which made the physical position of data on a track critical. This is called synchronizing. These disks were recorded with a special timing relationship between tracks, and in order to copy the disk correctly, it is necessary to copy them with the same timing relationship. Synchronizing tracks while copying accomplishes this.

When specifying synchronization, Locksmith will copy track 0 first. After analyzing the track, Locksmith will select a string of data to use as a synchronize pattern. When reading the next track (track 1 for instance) Locksmith will first seek to track 0, wait for the synchronize pattern to be read, and immediately seek to track 1 and start reading. This way, it knows the timing relationship between the two tracks. After analyzing the track it will be rewritten while preserving this relationship. Before writing, the head will seek to track 0, wait for the synchronize pattern, and step to track 1 and immediately start writing. This way the timing between the 2 tracks is the same on each disk. Normally, this is done on just one or two tracks on a disk, although it may be done on any number of tracks.

There is one other type of disk protection which requires synchronized copying. This is a disk which has spiral tracks. Normally, due to physical limitations of the read/write head on the disk drive, data may not be copied any less than 1 track apart on a disk. For example, if data was copied on track 1, 1.5, 2 and 2.5, the data that was written on each track would overlap onto the adjacent half track, rendering the track unreadable. When copying track 1.5, tracks 1 and 2 would be destroyed, when copying track 2, tracks 1.5 and 2.5 would be destroyed, etc. The only way to safely record data on all tracks and half tracks is to spiral the tracks.

Spiral tracks utilize adjacent tracks and half tracks. In order to do this, the timing between the tracks is critical. For example, we might wish to start spiraling at track 1. At track 1, we write data approximately 1/2 of the way around the disk. We then jump to track 1.5, write 1/2 of the way around the disk, jump to track 2, etc. By using only a portion of the available track, we can actually spiral the written data on the tracks. By the time we have completed one revolution of the disk, we are already on track 2. This preserves the 1 track separation while utilizing all tracks and half tracks. In order to back up this disk, it will be necessary to copy the small sections of data in synchronized mode. This must be done to insure that no data overlaps the adjacent half track. There are several other problems associated with spiral tracks, and these will be discussed in a future article.

There are several parameters which are used to modify the synchronize mode. These may be used to determine the synchronize type, the synchronize track, and the algorithm to use to determine the sync pattern. Unfortunately, there is no easy way to determine if synchronize mode is required for a disk. It is usually necessary to analyze the disk that is booting to determine if it is synchronized. The only other option is to try synchronizing a disk if it will not boot after a normal copy.

USERS MANUAL ERRATA

MAKE THE FOLLOWING CHANGES TO YOUR LOCKSMITH 5.0 USERS MANUAL:

The numbers below (6:5:6, for example) represent the Page:Paragraph:Line

6:5:6 - CHANGE 'I' TO 'IT'
25:6:1 - 'FIELD' TO 'FAILED'
25:1:1 - 'O' TO '0'
89:4:1 - 'SYNCPAT' TO 'SYPAT'
89:10:1 - 'SYNCTYP' TO 'SYTYP'
89:11:1 - 'SYNCTRK' TO 'SYTRK'
89:12:1 - DELETE THIS LINE
90:1 - DELETE THIS PARAGRAPH
90:5 - DELETE THIS PARAGRAPH
102 (LINE BEGINNING 'PARMS USED') -
CHANGE 'RANGE' TO 'SPAN'
105 (LINE BEGINNING 'PARMS USED') -
'VALUE' TO 'PARM 167'
110 (LINE BEGINNING 'PASS VALUE') -
DELETE 'DEFAULT=1'
110 (LAST LINE OF ALG # 26)
CHANGE 'IS THE DEFAULT VALUE' TO
'IS USED AS ONE OF THE DEFAULT
ALGORITHMS'
128 - THE FOLLOWING OPTIONAL KEYWORDS
CAN APPEAR ON THE COPY COMMAND:
SYNC=NO/YES
COUNT=NO/AUTO/MANUAL
MANERR=NO/YES

** DISK BACKUP SEMINAR **

A Disk Backup Seminar is being planned for several cities across the United States. The seminar will be reasonably priced and will be divided into two categories: beginner and advanced. The seminar will be taught by the author of Locksmith and other members of the Locksmith programming team, as well as other prominent individuals in the field of disk backup. To keep the seminar on a highly interactive level, the number of attendees in each will be limited. If you are interested in receiving further information regarding the seminar, please send your name and address to the following address:

**LOCKSMITH DISK BACKUP SEMINARS
P.O. BOX 424
UNION, ILLINOIS 60180**

When the dates and locations of the seminar determined, you will be sent the appropriate registration forms.

Renewal Form

Note-- This first issue of the Newsletter is being sent free of charge to all Locksmith 5.0 registered owners. You may subscribe to future issues, which will be published quarterly, or more-often, if the need arises. To receive future issues, send check or money order to:

Locksmith Newsletter
177 W. Hintz Rd.
Wheeling, Il. 60090

Subscription rates:

Locksmith Newsletter: \$12/year

Parameters on diskette: \$8/disk

Visa and Master Card accepted. No purchase orders!

Make all checks payable to Omega MicroWare, Inc..

Name _____

Address _____

City _____ St _____ Zip _____

Country _____

MC - Visa _____

Expiration date _____

Please enter my subscription to the following:

Locksmith newsletter / / \$12.00 (foreign \$18.00)

Parms diskette / / \$8.00 (foreign \$14.00)
(All known parameters on diskette)

PARMS

```
*ADVANCED VISICALC APPLE III
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES
```

```
*AIRSIM-1
SET PAT1 D5 AA ? FF /
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES
*WRITE PROTECT BEFORE RUNNING
```

```
*AKALBETH
SET PAT1 DD AA /
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 18 1 SYNC=YES
```

```
*ALIEN RAIN OR ALIEN TYPHOON
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 E 1 SYNC=YES
```

★APPLE LOGO
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 1 1 1 COUNT=YES

★APPLEOIDS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3.5 3.5 1
COPY 21.5 21.5 1

★APPLE PANIC
SET PAT1 DD AA /
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 C 1 SYNC=YES

★APPLE III BUSINESS GRAPHICS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★APPLE PRINT USING
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1 SYNC=YES

★APPLE WORLD
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1

★APPLE WRITER III
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★AUTOBAHN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 4 6 1 SYNC=YES
COPY 9.5 C.5 1 SYNC=YES

★A2 PB1 PINBALL
READ
CHANGE SS DHDR
SET PAT2 DB AB BF /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 15 1

★ACE WRITER
READ
CHANGE SS DHDR
SET PAT2 D5 AA 96 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

★ADDRESS BOOK
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★ADVENTURE TO ATLANTIS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★AIR TRAFFIC CONTROLLER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1

★ALGEBRA SERIES
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★APPLE ADVENTURE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★APPLE IIE BUSINESS GRAPHICS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★ARSENE LARCIN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1

★AZTEC
READ
CHANGE SS DHDR
SET PAT2 D5 AA 96 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★APPLEWRITER PREBOOT
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

★BANK STREET WRITER
SET PAT2 A5 96 BF
SET PAT3 (D5) (AB)
READ
CHANGE FRAME1
CHANGE SS DHDR
CHANGE PAT3 SS
TSTART PAT2 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
SET DRIVE 1,2
COPY 0 TO 1A BY 1
COPY 1B TO 22 BY 1 COUNT=AUTO

★BANDITS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1
COPY 1.5 1A.5 1
COPY 1C.5 1F.5 1 SYNC=YES

★BEER RUN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 D.5 1 SYNC=YES

★BLADE OF BLACKPOOL
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1
COPY 1.5 F.5 1
COPY 11 22 1 SYNC=YES

★BORG
SET PAT 1 DD AD DA /
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 B.5 1 SYNC=YES
COPY 0D 20 1 SYNC=YES

*BPI BUSINESS ACCTING SYSTEM
SET PAT1 D5 AA /
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*BRAIN SURGEON
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 4 4 1 COUNT=AUTO

*BUG ATTACK
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 13 1
COPY 1E 1E 1 COUNT=AUTO

*BAG OF TRICKS
READ
CHANGE SS DHDR
SET PAT3 DF /
CHANGE PAT3 SS
SET PAT2 D6 AA /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 15 1

*BASIC FRANCAIS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1

*BATTLE OF SHILO
READ
CHANGE SS DHDR
SET PAT2 D4 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*BOMB ALLEY
READ
CHANGE SS DHDR
SET PAT2 D4 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*BILL BUDGES 3D SYSTEM
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 8 1
COPY 11 12 1
COPY 15 17 1

*CHOPLIFTER
SET DATAFB 1
READ
CHANGE EXTEND BY 2
CHANGE SS DHDR
CHANGE SS TO NORM
CHANGE NORM TO SS
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 TO B BY 1
COPY C.25 TO 1E.25 BY 1
SET PAT1 FD D4 FD /
READ
CHANGE EXTEND 2
TSTART PAT1
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
TSTART BSPACE FF
LABEL A:WRITE STAT 1
IF SUCC 1 THEN GOTO B
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO C
CLEAR ERROR : GOTO 1
LABEL C
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO D
CLEAR ERROR : GOTO 1
LABEL D: * SHORTENING FAILED
LABEL B: * WRITE SUCCEEDED
COPY 20 TO 20 BY 1

*CANNON BALL BLITZ
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3 F 1 COUNT=AUTO

*CASTLE OF DARKNESS
SET PAT2 AB AB /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1 SYNC=YES

*CASTLE WOLFENSTEIN
SET FRAME1
SET PAT2 D5 AA B5 /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*COMPUTER FOOSBALL
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 9.5 1 SYNC=YES
COPY A A 1 SYNC=YES

*CONGLOMERATES COLLIDE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 1B 1B 1 COUNT=AUTO

*CRANSTON MANOR
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 18 18 1 COUNT=AUTO

*CROSSFIRE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 1 23 22 COUNT=AUTO

*CROSSWORD MAGIC
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*CRUNCH CRUMBLE AND CHOMP
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*CYBERSTRIKE
SET PAT2 D5 AA B5 /
SET PAT3 D5 AA F5 /
SET FRAME1
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART PAT3 STAT 3
IF FAIL 3 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 3 B 1 SYNC=YES
COPY 11 1 C 1 SYNC=YES

*CYCLOD
SET PAT2 D5 FF /
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES COUNT=AUTO
COPY 1.5 E.5 1 SYNC=YES
COPY 1F 20 1

*CYBORG
SET PAT 2 D5 AA FF /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*CDEX TRAINING PROGRAM
READ
CHANGE SS DHDR
SET PAT2 D6 AB 96 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*CARAIBES
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1

*CEILING ZERO
READ
CHANGE SS DHDR
SET PAT D6 AA /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 11 1

*COMPUTER AMBUSH
READ
CHANGE SS DHDR
SET PAT2 D4 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*COMPUTER BASEBALL
READ
CHANGE SS DHDR
SET PAT2 D4 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*COMPUTER QUARTERBACK
READ
CHANGE SS DHDR
SET PAT2 D4 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*DROL
SET DATAFB 1
SET VERLEN 10
SET DATAFB 1
SET MAXLEN 10
READ
CHANGE NORM TO SS
TSTART LONG SS
CHANGE SS INVALIDS TO NORM
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND BSPACE 20
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 11
COPY 2.25 4 1.75
SET PARM 167 7F
SET PAT1 D7 DA /
READ
TSTART PAT1
TSTART BSPACE 1
TSTART ASSIGN PAT1
SET PAT1 + 2 ? /
TEND PAT1
TEND ASSIGN PAT2
SET PAT2 + 9 00 ...
CHANGE PAT2 TO VALUE
TEND REPEAT
DISPLAY
TSTART BSPACE FF
TEND BSPACE 10
VSTART PAT1
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6

IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 5 22 1

*DEFAULT LOCKSMITH LPL FILE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*CRISIS MOUNTAIN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*CONGO
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*COPTS AND ROBBERS
READ
CHANGE SS DHDR
SET PAT2 DD AD DA /
TSATRT PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1 5 F 5 1 SYNC=YES

*DARK FOREST
SET PAT2 D5 AA AF /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*DATADEX
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 2 1 SYNC=YES
COPY 3 5 3 5 1 SYNC=YES
COPY 5 22 1 SYNC=YES

*DATA PLAN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*DB MASTER V4.0
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 5 1 SYNC=YES
COPY 6 5 22 5 1 SYNC=YES
*WRITE PROTECT COPY BEFORE RUNNING

*DISK LIBRARY
SET PAT1 D5 AA 96 AA AA /
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
LEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*DISK ORGANIZER II
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 2 B 1 SYNC=YES
COPY 1 1 1 COUNT=AUTO

*DISK RECOVERY
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 2 16 1 SYNC=YES

*DRAGON GAMES
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 4 4 1 COUNT=AUTO

*DATESTONES OF RYN
READ
CHANGE SS DHDR
SET PAT1 D5 AA 96 /
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*ESCAPE FROM RUNGISTAN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*EPOCH
SET PAT2 D5 AA DA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 1 1 SYNC=YES
COPY 1 5 F.5 1 SYNC=YES

*ESCAPE FROM ARCTURUS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*EXPEDITER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3 1F 1C COUNT=AUTO

*EZ DRAW 3.3
READ
CHANGE SS DHDR
SET PAT2 D7 AA 96 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*FREEFALL
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 07 1
COPY 11 11 1 COUNT=AUTO

*FS-1
SET PAT2 DB AB BF /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 00 1
COPY 1.5 21 1.5
COPY 7 8 1
COPY 9.5 9.5 1

*FLIGHT SIMULATOR II
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 022 1 SYNC=YES COUNT=AUTO

*FLIGHT SIMULATOR A2-FS1
SET PAT1 DB AB FB /
SET FRAME1
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 00 1
COPY 1.5 TO 21 BY 1.5
*WRITE PROTECT BEFORE RUNNING

*FALCONS
SET PAT2 DF AD /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 00 1 SYNC=YES
COPY 1.5 4.5 1.5 SYNC=YES
COPY 5.5 5.5 1 SYNC=YES
COPY 7 A 1 SYNC=YES
COPY B.5 E.5 1.5 SYNC=YES
COPY 10 12 1 SYNC=YES
COPY 13.5 14.5 1 SYNC=YES
COPY 16 19 1.5 SYNC=YES
COPY 1A 1B.5 1.5 SYNC=YES

*FINANCIAL CONTROLLER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD

SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 022 1 SYNC=YES

*FIREBIRD
SET PAT2 DD AD DA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 00 1 SYNC=YES
COPY 1.5 B.5 1 SYNC=YES

*FACEMAKER
READ
CHANGE SS DHDR
SET PAT2 (2A) (1B) (FC) /
CHANGE PAT2 SS
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*GALACTIC SAGAS I AND II
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1 SYNC=YES

*GENETIC DRIFT
SET PAT2 BB B5 BB /
SET PAT3 D4 D5 BB /
SET PAT4 AD B5 DE /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART PAT3 STAT 3
IF FAIL 3 THEN TSTART PAT4 STAT 4
IF FAIL 4 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 3 1
COPY 4.5 6 1.5
COPY 7.5 B.5 1
COPY D D 1
COPY E.5 12.5 1

*GALATIC SAGA IV
SET LEADFB 1
SET DATAFB1
SET PAT2 D5 AA B5 /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*GAMMA GOBLINS
SET PAT2 DD AD DA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED

LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 D.5 1 SYNC=YES

*GOBBLER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3 3 1 COUNT=AUTO

*GOLDEN MOUNTAIN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1 D 2 SYNC=YES
COPY 2 E 2

*GORGON
SET PAT2 DD AD DA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 E.5 1 SYNC=YES

*GUADALCANAL CAMPAIGN
READ
CHANGE SS DHDR
SET PAT3 (6E) (FE) /
CHANGE PAT3 SS
SET PAT2 D4 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED

LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*GALACTIC GLADIATORS
READ
CHANGE INVALID SS
CHANGE SS DHDR
SET PAT2 D7 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*GALACTIC ATTACK
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*HADRON
SET PAT2 DD AD DA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 E.5 1 SYNC=YES

*HAYDEN ALIBI
SET PAT2 D4 B5 /
READ
CHANGE SS DHDR
TSTART PAT STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED

COPY 0 22 1
COPY 1B 1B 1 COUNT=AUTO

*HAYDEN APPLESOFT COMPILER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*HIRES CRIBBAGE AND FOOTBALL
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 5 1 SYNC=YES
COPY 6 22 1

*HIRES GOLF
SET DATAFB1
SET LEADFB1
SET PAT2 D5 AA B5 /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*HIRES SOCCER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*IMAGE PRINTER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 7 1 SYNC=YES
COPY 9 22 1 SYNC=YES
COPY 8 8 1 COUNT=AUTO

*INCREDIBLE JACK
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
*WRITE PROTECT BEFORE USING

*HYPER HEAD ON
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 12 2 SYNC=YES

*HOMEWORD
SET DATAFB 1
SET PAT1 (D5) (AA) (96) /
READ
CHANGE SS DHDR
CHANGE INVALIDS TO 7F
CHANGE NORM TO SS
CHANGE PAT1 TO NORM
CHANGE SS INVALIDS TO NORM
TSTART LONG SS
ANCHOR
TSTART BSPACE
TSTART PAT1
TEND REPEAT
DISPLAY
TSTART BSPACE
VSTART NORM
TEND FSPACE 3
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1

LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 COUNT=AUTO

*JAWBREAKER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3 3 1 COUNT=AUTO

*JELLYFISH
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1 5 F.5 1 SYNC=YES
COPY 11 11 1 COUNT=AUTO

*JIGSAW
READ
CHANGE SS DHDR
SET PAT2 ? AA 96 ? /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 17 1 SYNC=YES

*KABUL SPY
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES COUNT=AUTO
COPY 1.5 10.5 1 SYNC=YES
COPY 11 11 1 COUNT=AUTO
COPY 14 20 1
COPY 23 23 1 COUNT=AUTO

*LEMMINGS
SET PAT2 DD AD DA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 B.5 1 SYNC=YES
COPY D 20 1 SYNC=YES

*LETTER PERFECT
SET PAT2 ? D5 AA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*LIST HANDLER
READ
CHANGE SS DHDR
CHANGE SS TO NORM
CHANGE NORM TO SS
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 TO 11 BY 1
COPY 12.25 TO 22.25 BY 1

*LODE RUNNER
SET DATAFB 1
SET PAT1 D4 D5 D6 /
SET PAT2 (D5) (AB) /
READ
CHANGE PAT2 TO SS
CHANGE SS DHDR
TSTART PAT1
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED

LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
SET PAT1 DD F5 D5 /
COPY 21 21 1 SYNC=YES
SET PAT1 D5 AA 96 ? ? ? ? AA AA /
COPY 3 C 1 SYNC=NO
READ
TSTART LONG NORM
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY D.25 1F.25 1 SYNC=YES

*LE CHOMEUR
READ
CHANGE SS DHDR
SET PAT1 D5 AA 96 /
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1

*LIST HANDLER
READ
CHANGE SS DHDR
CHANGE SS TO NORM
CHANGE NORM TO SS
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 TO 11 BY 1
COPY 12.25 TO 22.25 BY 1

*MAD VENTURE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1 SYNC=YES

*MICRO BASEBALL
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*MICROWAVE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 11 11 1 COUNT=AUTO

*MILLIKEN MATH
SET DATAFB 1
SET LEADFB 1
SET PAT2 D5 AA B5 /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*MISSILE DEFENSE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*MISSION ASTERIOD
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*MOUSKATTACK
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 23 23 1 COUNT=AUTO

*MASTERTYPE
SET LEADFB 1
SET DATAFB 1
SET PAT2 D4 AA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 1A 1
COPY 1C 22 1

*MATH STRATEGY
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*MICROBE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*MINER 2049ER
READ
CHANGE SS DHDR
SET PAT2 D3 96 F2 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES COUNT=AUTO
COPY 1 22 1 SYNC=YES

*MISSING RING
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*MINER 2049'ER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*NIGHT MISSION
SET PAT2 BB AB DF /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 15 1

*OUTPOST
SET PAT2 DD AD DA
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5D.5 1 SYNC=YES

*OPERATION APOCALYPSE
READ
CHANGE SS DHDR
SET PAT2 DB D5 DE /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*ORBITRON
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 1 1
COPY 1.5 F.5 1
*WRITE PROTECT BEFORE RUNNING

*PADDLE GRAPHICS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 23 23 1 COUNT=AUTO

*PEGASUS II
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3 3 1 COUNT=AUTO

*OLYMPIC DECATHLON
SET PAT2 D5 AA B5 /
SET LEADFB 1
SET DATAFB 1
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*PFS GRAPH
SET PAT2 DB AB BF /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED

COPY 0 0 1
COPY 1.5 21 1.5
COPY 7 8 1
COPY 9.5 9.5 1
* MUST BE WRITE PROTECTED!!

*PFS (OLD VERSION)
SET PAT2 93 F3 FC FF /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 13 1
*MUST BE WRITE PROTECTED!

*PFS REPORTS (OLD VERSION)
SET PAT2 93 F3 FC FF /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 13 1
*MUST BE WRITE PROTECTED

*PHANTOMS FIVE
SET PAT2 DD AA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 1C 1

*POOL 1.5
SET PAT2 D5 AA B5 /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 15 1 SYNC=YES
COPY 1E 21 1 SYNC=YES

*PRESIDENT ELECT
READ
CHANGE SS DHDR
SET PAT2 D4 AA B7 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*PFS FILE
SET VERLEN TO C
READ
CHANGE SS DHDR
CHANGE INVALIDS TO (FF)
CHANGE NORMAL TO SS
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 4 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 4 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*PULSAR II
SET PAT2 DD AA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES

COPY 2C 1 SYNC=YES
COPY 13 19 1 SYNC=YES
COPY 1A.5 1D.5 1 SYNC=YES

*RINGS OF SATURN
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*RASTER BLASTER
SET VERLEN 8
SET LEADFB 1
SET DATAFB 1
SET PAT1 TO D5 AA B5 /
SET PAT2 TO AD DE /
READ
TSTART PAT1
VSTART PAT1
TEND REPEAT
TSTART BSPACE 10
WRITE
COPY TRACK 0 TO 0 BY 1
READ
TSTART PAT2
VSTART PAT2
TSTART BSPACE 10
TEND TSTART 16
WRITE
COPY TRACK 5 TO 11 BY 4
COPY TRACK 6 TO 12 BY 4
COPY TRACK 7.5 TO F.5 BY 4
COPY TRACK 1.5 TO 3.5 BY 2

*QUICK LOADER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 2 11 1 SYNC=YES

*RETROBALL
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1
COPY 4 6 1
COPY 9 C 1
COPY E 10 1

COPY 12 14 1
COPY 17 1D 1
COPY 20 22 1

*RENDEZVOUS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 23 1 SYNC=YES

*ROBOT WARS
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*RASTER BLASTER
SET VERLEN 8
SET LEADFB 1
SET DATAFB 1
SET PAT1 TO D5 AA B5 /
SET PAT2 TO AD DE /
READ
TSTART PAT1
VSTART PAT1
TEND REPEAT
TSTART BSPACE 10
WRITE
COPY TRACK 0 TO 0 BY 1
READ
TSTART PAT2
VSTART PAT2
TSTART BSPACE 10
TEND TSTART 16
WRITE
COPY TRACK 5 TO 11 BY 4
COPY TRACK 6 TO 12 BY 4
COPY TRACK 7.5 TO F.5 BY 4
COPY TRACK 1.5 TO 3.5 BY 2

*SABATOGUE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3 3 1 COUNT=AUTO

*SHATTERED ALLIANCE
SET PAT2 D4 AA B7 /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*SNAKE BYTE
SET PAT2 DD AD DA /
READ
CHANGE SS DHDR
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 A.5 1 SYNC=YES

*SNEAKERS
READ
CHANGE SS DHDR
SET PAT2 DD AD DA /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 C.5 1 SYNC=YES

*SNOGGLE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC
COPY 1 F 1
COPY 10.5 11.5 1 SYNC=YES

*SOFTPORN ADVENTURE
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 2 1
COPY 3 3 1 COUNT=AUTO

*SPACE EGGS
READ
CHANGE SS DHDR
SET PAT2 DD AA /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1
COPY 2 5 1
COPY 11 1A 1

*SPACE QUARKS
READ
CHANGE SS DHDR
SET PAT2 AB D4 AB /
SET PAT3 FE DD AF /
SET PAT4 AA DE BB /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART PAT3 STAT 3
IF FAIL 3 THEN TSTART PAT4 STAT 4
IF FAIL 4 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 2 1
COPY 3.5 5.5 1
COPY 7 9 2
COPY A.5 B.5 1
COPY D 15 1

*SPACE WARRIOR
READ
CHANGE SS DHDR
SET PAT2 DF AD DE /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD

CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1
COPY 2.5 3.5 1
COPY 5 8 3
COPY 6.5 6.5 1
COPY A 10 3

*STAR BLASTER
READ
CHANGE SS DHDR
SET PAT2 DF AD DE /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 7 20.5 1.5 SYNC=YES

*STAR CRUISER
READ
CHANGE SS DHDR
SET PAT2 AA DD BB /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 033 SYNC=YES
COPY 4B 1 SYNC=YES
COPY 11 12 1 SYNC=YES

*STAR MINES
READ
CHANGE SS DHDR
SET PAT2 D5 AA AD /
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED

COPY 02 1
COPY 4A 1

*SPECTRE
READ
CHANGE SS DHDR
SET PAT2 C5 AA B5 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 022 1

*STAR THIEF
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 013 1
COPY 22 22 1 COUNT=AUTO

*SPELLING STRATEGY
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 022 1 SYNC=YES

*SPITFIRE SIMULATOR
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0F 1
COPY 15 15 1

*STARSHIP COMMANDER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 022 1 SYNC=YES

*THIEF
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 00 1 SYNC=YES
COPY 13 1
COPY 45 1 SYNC=YES
COPY 622 1

★THRESHOLD
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1
COPY 1 1 1 COUNT=AUTO
COPY 2 2 1
COPY 22 22 1 COUNT=AUTO

★TINY TROL
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1 2 2 1
COPY 3.5 5 1.5 SYNC=YES

★TWERPS
READ
CHANGE SS DHDR
SET PAT 2 DD AD DA /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 SYNC=YES
COPY 1.5 E.5 1 SYNC=YES
COPY 1C 1C 1 SYNC=YES COUNT=AUTO

★TAWALAS LAST REDOUBT
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

★THUNDERBOMBS
READ
CHANGE SS DHDR
SET PAT2 ? AA 96 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 12 1

★TRANSYLVANIA
READ
CHANGE SS DHDR
SET PAT2 ? AA 96 /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

★TORPEDO FIRE
READ
CHANGE SS DHDR
SET PAT2 D4 AA 96 /
CHANGE INVALID SS
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

★TYPE ATTACK
READ
CHANGE SS DHDR
SET PAT2 AD DA DD /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 E 1 SYNC=YES

★U BOAT COMMAND
 READ
 CHANGE SS DHDR
 SET PAT2 ?? EB AF /
 TSTART PAT2 STAT 2
 IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 22 1

★ULYSIS
 READ
 CHANGE SS DHDR
 TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 22 1
 COPY 2 2 1 COUNT=AUTO

★UDRAW II
 READ
 CHANGE SS DHDR
 TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 22 1 SYNC=YES

★ULTIMA III
 SET DATAFB 1
 READ
 CHANGE SS DHDR
 CHANGE SS INVALID TO NORM
 TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 10 1

★VISIDEX
 READ
 CHANGE SS DHDR
 SET PAT2 AA EB FD /
 TSTART PAT2 STAT 2
 IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 22 1

★VISISCHEDULE
 READ
 CHANGE SS DHDR
 SET PAT2 AA EB EC /
 TSTART PAT2 STAT 2
 IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 22 1

★VISITERM
 READ
 CHANGE SS DHDR
 SET PAT2 AA EB FC /
 TSTART PAT2 STAT 2
 IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 22 1

★VISIFILE
 READ
 CHANGE SS DHDR
 SET PAT3 (2A) /
 CHANGE PAT3 SS
 TSTART DOS PAT1 STAT 1
 IF FAIL 1 THEN TSTART LONG SS
 TEND REPEAT
 DISPLAY
 TSTART BSPACE
 TEND FSPACE 3
 VSTART NORM
 LABEL 1:WRITE STAT 1
 IF SUCC 1 THEN GOTO EE
 SHORTEN ALL EQUAL BY 2 STAT 5
 IF FAIL 5 THEN GOTO DD
 CLEAR ERROR : GOTO 1
 LABEL DD
 SHORTEN ALL CENTER BY 2 STAT 6
 IF FAIL 6 GOTO DA
 CLEAR ERROR : GOTO 1
 LABEL DA: * SHORTENING FAILED
 LABEL EE: * WRITE SUCCEEDED
 COPY 0 22 1

*WIZARDRY
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
TSTART BSPACE
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 0 1 COUNT=MANUAL SYNC=YES
COPY 1 22 1 SYNC=YES

*WORD HANDLER
READ
CHANGE SS DHDR
SET PAT2 FF DF DE /
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 C 1
COPY 11 22 1

*WARP FACTOR
READ
CHANGE SS DHDR
SET PAT2 DB D5 DE /
TSTART PAT2 STAT 2
IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
*WRITE PROTECT BEFORE USING

*WIZ MAKER
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*ZORK I.II.III
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1
COPY 3 3 1 COUNT=AUTO

*ZARDAX
READ
CHANGE SS DHDR
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1 SYNC=YES

*ZOOM GRAPHICS
READ
CHANGE SS DHDR
SET PAT2 D4 AA 96 /
TSTART PAT2 STAT 2

IF FAIL 2 THEN TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 22 1

*ZAXXON
*NOTE:SEVERAL VERSIONS OF ZAXXON
* SEE BOTTOM OF THIS FILE
SET DATAFB 1
SET PARM 01B TO 80 80
READ
CHANGE SS DHDR
CHANGE SS INVALIDS TO NORMAL
TSTART DOS PAT1 STAT 1
IF FAIL 1 THEN TSTART LONG SS
TEND REPEAT
DISPLAY
TSTART BSPACE
TEND FSPACE 3
VSTART NORM
LABEL 1:WRITE STAT 1
IF SUCC 1 THEN GOTO EE
SHORTEN ALL EQUAL BY 2 STAT 5
IF FAIL 5 THEN GOTO DD
CLEAR ERROR : GOTO 1
LABEL DD
SHORTEN ALL CENTER BY 2 STAT 6
IF FAIL 6 GOTO DA
CLEAR ERROR : GOTO 1
LABEL DA: * SHORTENING FAILED
LABEL EE: * WRITE SUCCEEDED
COPY 0 13 13 COUNT=AUTO
COPY 1 12 1
*NOTE:IF ABOVE DOESN'T WORK, TRY
* COPY 0 0 BY 1 COUNT=AUTO
* COPY TRACK 1 TO 16 BY 1
* COPY TRACK 20 TO 20 BY 1