

**DENSE-PAC**  
**DATA COMPRESSION AND ENCRYPTION PROGRAM**  
*Copyright 1983 by D. Wesson*  
*Produced by Newline Software*

The **DENSE-PAC** Utility program was designed to help reduce disk space required for storage of text files, and to provide you with security of your text via password protection. Text files you may wish to compress include: correspondence, reports, program source code, or even documentation. Files of this type use a considerable amount of space. Using **DENSE-PAC**, you will be able to recover a significant amount of disk space and still keep all your important information "on-line." The files will also be encrypted, providing a level of security for your sensitive or private data.

**EASE OF USE**

Complete 'cookbook' documentation for ease of operation. All keyboard entries by the operator fully explained in the program make the program very 'user friendly'.

**SPEED**

Fast compression and encryption — archives better than 512 characters per second. Can archive a 90 sector file in 39 seconds and restore it in 37 seconds.

**EFFICIENCY**

Archives ASCII text files with an average compression of 30% to 40% or more, reducing backup/storage disks by 30% or more. Capable of archiving any file — including non-ASCII files such as machine language files. Uses modified Huffman code compression along with keyword compression for the best compression available. Not limited by memory size or number of disk drives—will archive or restore any size file that can fit on your disk.

**SECURITY**

Archived files are not readable by any editor thereby increasing security. Optional passwords further increase security of archived files. Files are non-restorable without the correct password. Passwords allow entry of ANY key on the keyboard that is sent to the computer, increasing the total passwords available. Ability to 'archive' a previously archived file increases security of files — no limit on the number of times a file can be archived with a different password each time.

**VERSATILE**

Single drive operation fully supported — including swapping and mounting disks.

For CP/M 2.x, H8, H/Z-89 . . . . . **\$29.95**

**DATA COMPRESSION**

**\$29.95**

**DENSE-PAC DATA  
 COMPRESSION  
 AND ENCRYPTION  
 UTILITY**

A program to compress data files and provide a degree of security for your data. Transmit compressed files via modem and save money on long distance connects for your important data space requirements for your important data and programs.

Copyright 1983 by D. Wesson  
 Produced by Newline Software  
 ALL RIGHTS RESERVED

**Requires:**

H8 or H/Z-89 Computer System  
 One or more disk drives  
 CP/M 2.x

\*H8, H/Z-89 are trademarks of Heath Company  
 CP/M is a trademark of Digital Research, Inc.

DENSE-PAC UTILITY

Users Guide and Reference Manual  
Copyright 1983 by D - Soft  
Produced by Newline Software

CP/M VERSION  
12 July 1983

All rights reserved - supplied  
for single end user use unless  
specified in an agreement to  
the contrary.

HEATH is a registered trade mark of the HEATH CO.  
CP/M is a registered trade mark of Digital Research



## TABLE OF CONTENTS

	Page
1.0 At A Glance	1
2.0 Introduction	2
3.0 Description	4
3.1 Distribution disk	5
3.2 Main menu	5
3.3 Archive function	6
3.3.1 Introduction	6
3.3.2 Archive types	6
3.3.3 Archive parameters	7
3.3.3.1 Case conversion	8
3.3.3.2 End Of File (EOF)	8
3.3.3.3 Passwords	9
3.3.3.4 Keywords	9
3.3.3.5 Quotes	10
3.3.4 Input / output files	10
3.3.5 Statistics	11
3.3.6 An archive trip	12
3.4 Find function	15
3.4.1 Introduction	15
3.4.2 Find parameters	15
3.4.2.1 Case conversion	16
3.4.2.2 End Of File (EOF)	16
3.4.2.3 Keywords	16
3.4.2.4 Quotes Strings	16
3.4.3 Input file	16
3.4.4 Statistics	17
3.4.5 An find trip	17
3.5 Restore function	21
3.5.1 Introduction	21
3.5.2 Input / output files	21
3.5.3 Statistics	22
3.5.4 An restore trip	22
3.6 Archive / verify function	25
3.6.1 Introduction	25
3.6.2 Verify outputs	26
3.6.3 Verify statistics	27
3.6.4 An archive / verify trip	27
3.7 Swap function	33
3.7.1 Introduction	33
3.8 Oops errors	33
3.8.1 Introduction	33
3.8.2 Errors explained	34
3.8.2.1 Operator errors and warning messages	34
3.8.2.2 Program detected hardware errors	35
3.8.2.3 Program detected software errors	35
Appendix A - Delivered keywords	37



## 1.0 AT A GLANCE

### EASE OF USE

Complete 'cookbook' documentation for ease of operation.

All keyboard entries by the operator fully explained in the program make the program very 'user friendly'.

### SPEED

Fastest archive utility available - archives better than 450 characters per second. Can archive a 20 Kbyte file in 44 seconds and restore it in 42 seconds.

### EFFICIENT

Archives ASCII text files with an average compression of 30 to 40 % or more, reducing backup/storage disks by 30% or more.

Capable of archiving any file - including non - ASCII files such as machine language files.

Uses modified Huffman code compression along with keyword compression for the best compression available.

Not limited by memory size or number of disk drives - will archive or restore any size file that can fit on your disk.

### SECURITY

Archived files are not readable by any editor increasing security.

Optional passwords increase security of archived files. Files are non - restoreable without the correct password.

Passwords allow entry of ANY key on the keyboard that is sent to the computer, increasing the total passwords available.

Ability to 'archive' a previously archived file increases security of files - no limit on the number of times a file can be archived with a different password each time.

### VERSATILE

Single drive operation fully supported - including swapping disks.

Five functions available :

1. Archive a file
2. Archive a file and then verify it is correct
3. Find the best algorithm to archive a file with
4. Restore a file
5. Swap disk

Ability to view a file while it is being verified or restored on the terminal screen.

Five complete algorithms designed for several text types :

1. Upper and lower case text
2. Upper case text
3. Assembly language source code
4. Pascal language source code
5. Basic language source code

Selectable archive parameters enhance compression.

Complete statistical data after each function.

Ability to abort the current operation.

## 2.0 INTRODUCTION

The DENSE-PAC Utility program was designed to help you reduce the disk space required for storage of text files, and to provide you with security of your text via password protection. When we speak of text files, we are referring to many types of files that you use. These files may be correspondence, reports, program source code, or even documentation such as this. Files of this type are normally what you would like to archive and use a considerable amount of space due to their storage technique. Usually, a text file will use one byte (8 binary bits) of storage for each ASCII character in the file but ASCII characters require only 7 of these bits. By eliminating just one bit, a 12.5% reduction or 'compression' of storage space used will be realized. Unfortunately single bit compression yields a limited amount of savings while better methods are available. A note on terminology is in order. When we refer to compression we mean that the text is 'compressed' into a smaller length. De-compression is the opposite, restoring the compressed form into the readable form.

If you were to study the contents of a typical text file, you would note that a great amount of the file is devoted to common ASCII characters. For example when we studied the contents of typical text files, we found that spaces can consist of up to 27% of the file. Naturally this figure is dependent on the type of text file and will vary from one file to another. Taking into account that backup storage of text files take a great deal of disk space not to mention several disks, we decided a better more compact form of archiving was needed. Capitalizing on the fact that common ASCII characters could be represented with shorter bit lengths than 8, the DENSE-PAC Utility program was born.

Using modified Huffman code algorithms combined with keyword detection we can compress a text file into a serial bit stream. The DENSE-PAC Utility program can reduce the size of a text file up to 67% (using ideal test cases). On the average, we have found that we can compress a typical text file 30% to 40%, thus reducing your disk space used and make your archiving tasks easier. The serial bit stream method 'compresses' the text by substituting a binary bit stream for each character. The binary

bit stream is variable length dependent on the character and the number of times it occurs. The bit streams can vary from 2 bits up to 13 bits long and are compressed by our special algorithms. For example if we assign each of the following characters a special bit stream we could compress the word 'HEATH' from 40 bits to 22 bits for a 45% savings.

Character	Bit stream
A	101
E	1000
H	00111
T	11001

ASCII bit stream

01001000	01000101	01000001	01010100	01001000
H	E	A	T	H

DENSE-PAC Utility bit stream

00111	1000	101	11001	00111
H	E	A	T	H

Note that this example uses fairly common letters that are repeated quite often in text. If we were to take a word with less commonly used letters (such as Z, X, Q, etc) we could possibly exceed the number of bits that ASCII takes to encode the word. The increased length is due to the fact that we assign longer bit streams to less used characters.

The beauty of the system is that we assign short bit streams to the common letters and longer ones to the less common letters, thus getting a good overall compression of the text. Ideally it would be nice if all programs could use this method of text storage, but the processing time would be too great. The example above does not contain actual bit streams used in the DENSE-PAC Utility but the lengths of the bit streams are some of the lengths we use.

Our Huffman code algorithms were designed by counting the number of times a character appears within a text file. Using the total number of characters within the text file and comparing it to the number of times a character is present, we can figure out the frequency of occurrence of any individual character. The type of text sampled plays a very important part in building the frequency of occurrence tables. Since types of text can vary such as a small child's book compared to a highly technical book, we took special care to get a representative sample of text that was not biased toward any level of difficulty. When comparing our final sample to the text samples, we found the frequency of occurrence of individual characters differ very slightly. To get a valid sample we have taken text from several different sources and sampled over 80,000 characters of text to build our algorithm. While building our algorithm we discovered that while English text was basically similar in frequency of occurrences, computer languages were not. There for we also designed algorithms for the Assembly, BASIC and Pascal languages. All together, we sampled over 290,000 characters of various types of text to build our algorithms. We feel they are very complete.



In every language, English or computer, there are certain words or roots of words that tend to occur rather frequently. In computer languages these are called 'reserved' or 'keywords' such as BASIC's GOTO statement. To enhance compression, the DENSE-PAC Utility will search for these keywords and compress them into a special code, giving better compression. Refer to appendix A for a list of the keywords used for each algorithm. In our example above, we compressed the word 'HEATH' into a serial bit stream of 22 bits. If we were to take the keyword 'EAT' out of the word and use a special code say 7 bits long to represent the keyword, we would reduce the 22 bits to 17 bits increasing our compression to 57.5%.

By using separate algorithms for the type of text combined with keyword detection you get a reliable, fast and efficient way to reduce the size of your text files. The size of a text file is not limited to any size contrary to some 'compression' type programs, the DENSE-PAC Utility can reduce the size of a file as small as 2 Kbytes.

Since the archived output of the DENSE-PAC Utility is a serial bit stream of data, the output file is reduced to an unreadable form which increases security of sensitive files. For example taking the above example of 'HEATH', the archived output would read '<\?' where the '?' is a non-printable character in the ASCII code. To further enhance the security, DENSE-PAC Utility allows a password to be included with the compressed file. This password is also compressed into the file making de-compression of the file (restoring) very improbable unless the proper password is known. Of course, the DENSE-PAC Utility is needed to restore any archived file.

To enhance the reliability of the DENSE-PAC Utility, a verify function is included that will verify that the archived output is correct and can be restored. This is done by actually restoring the file and comparing it to the original file thus insuring the validity of the archived file. You can also view the file on the terminal as it is being verified or restored.

### 3.0 DESCRIPTION

The DENSE-PAC Utility has two primary functions, archiving a text file and restoring it to the original form. Along with the primary functions secondary functions such as verify, finding the best algorithm to use, and swapping of disks are included.

When you call the DENSE-PAC Utility, it will find the bottom of BDOS and divide the remaining memory into two buffers. The buffer size will depend upon the amount of remaining memory but the DENSE-PAC Utility will run with as little as 1024 bytes of memory for buffers. When your amount of memory is limited, the DENSE-PAC Utility will run slower due to more disk reads and writes. Using this buffer concept allows the program to compress any size of program that you may have. The only restriction is that if you do exceed 65K of input bytes,

the information counters will overflow. This does not affect the operation of the program at all and the informational printouts will be X \* 65k off where X is the number of times the counts overflowed.

After initialization is completed, you will be asked to select the function you wish to do. Each function is driven separately and will return to this menu when its task is completed, thus allowing you to select another function. It is possible to archive all of your text files at one sitting without reloading the program or restarting. Facilities are also included to abort the current function via a Control - B at most times during the program and return to the main menu. This is the only way to correct any input that you have made by mistake such as requesting archiving when you really wanted archiving with verify. You may also abort the entire program at most times via a Control - C, which will return you to CP/M.

All of your entries are tested for correctness and you will be informed if an entry is incorrect and requested to re-enter the data. Considerable time was invested in the design of the program to make the entries understandable and concise for you. We tried to keep the amount of information entered to a minimum by using default values wherever practical. For example, whenever you are asked to answer a question with a YES / NO answer (depicted by '< Y/N >'), you only have to enter 'Y', 'y' or a carriage RETURN for YES or a 'N', 'n' for a NO. When file names are required from you, we have tried to use default values when ever possible relieving you of typing the standard file parameters. The carriage RETURN is only required to terminate an entry when entering file names, it can also be used as the default on YES - NO questions. While the archive, restore, find or archive with verify function is executing, the 25 th line on the display will contain the information about the files that DENSE-PAC Utility is working with.

### 3.1 DISTRIBUTION DISK

The distribution disk contains the following three files.

DENSEPAC.COM	The archive program.
DOCFIELD.DOC	A file of text to use when taking DENSE-PAC Utility trips.
DOC.ARK	An archived file of text to use when taking DENSE-PAC Utility trips.

### 3.2 MAIN MENU

The main menu is the complete list of all the main functions available to you while executing the DENSE-PAC Utility. All of the selections will return to the main menu, allowing you to pick your next selection. Listed below is the main menu and the paragraphs describing the functions.

Selection Character	Function	Paragraph
A =	Archive file	3.3
V =	Archive and verify file	3.6
F =	Find best archive algorithm	3.4
R =	Restore file	3.5
S =	Swap disks	3.7
E =	Exit program	Below

The exit function will clear the screen, clean up any files and exit to CP/M.

### 3.3 ARCHIVE FUNCTION

#### 3.3.1 INTRODUCTION

The archive function is used to compress your text files and write out the compressed version on your disk. To select the archive function, enter a 'A' as your main menu selection.

#### 3.3.2 ARCHIVE TYPES

The first question you will be asked to answer is the type of text you wish to archive. This selection of type determines which of the archive algorithms to use to compress your text and is very important in getting a good compression ratio. Listed below is the selections available :

Type #	Description
0	= Upper and lower case text
1	= Upper case text
2	= Assembler source code
3	= Pascal source code
4	= Basic source code

The type of text is also important when determining the archive parameters as explained in paragraph 3.3.3. As a general rule of thumb; if your text is source code for a program language, use that type to archive your program. Remember that it is not only the keywords that makes the individual algorithm more efficient. Most computer languages use punctuation for special meanings, such as the ';' in Pascal. These were also taken into consideration when the algorithms were designed. If you are using another computer language such as FORTRAN, FORTH, etc; you can pick a language type that is similar or you can use one of the text types.

Type '0' for upper and lower case text is the only type that automatically archives lower case text. All of the other types will automatically convert lower case characters to upper case characters unless directed otherwise. See paragraph 3.3.3.1 for a further explanation of case conversion. Since type '0' includes both upper and lower case text, this type is the best one for text like you are reading right now. All of the algorithms are capable of compressing both cases of text, it's

just that type '0' does it more efficiently than the others do. On the other hand, if your text is primarily upper case, you would want to use type '1'.

If you have a file (not necessarily text) that does not seem to fit any of the descriptions, we recommend that you use type '0'. Type '0' has the most versatile algorithm and is well suited for general purpose archiving. If you are totally at a loss as to which type to use, you can use the find function to find the best algorithm for your text. The find function is explained in paragraph 3.4.

One thing to remember is that all of the algorithms are based on ASCII text files that contain only ASCII characters. If you wish to compress a file that contains assembled code (COM file), the archive function will do it, but since an assembled code file contains values that are not in the range of ASCII code ( > 127 ) you will probably find that the resultant output archived file is larger than the input! This is because all values that are out of the ASCII range are archived as special cases and take longer bit streams. In this case, the only gain to be realized is security. Remember that archive files are unreadable and cannot be executed!

One final note, if you play around with the archive function, you may find that your text file will 'compress' more efficiently with one type than another (we certainly hope so). You may also find that the text file compresses better with the 'wrong' type. That is, a text file compresses better with say the Pascal type. Don't panic! Nothing is wrong. It just happens that your text file 'looks' better to the Pascal algorithm than to the text algorithm. This is because it probably uses characters that are frequently found in Pascal text and not so frequently found in normal text. This occurrence is not an error. We have tried to 'customize' the algorithms as much as possible but a certain amount of overlap will happen. Be happy that you found a better type.

### 3.3.3 ARCHIVE PARAMETERS

After you enter your text type, you will be asked to verify various parameters that the archive function will use. These parameters are very important for you because they describe your text for the archive function. The parameters are listed below:

Buffer size	Convert lower to UPPER
Password protection	Normal End Of File
Keyword searching	Quote delimiter

When the parameters are displayed, the current settings will be displayed also. The buffer size is informational only and cannot be changed. This size is the amount of memory available for one buffer (remember that there are 2 buffers). The next 4 parameters will have a 'Y' or a 'N' displayed to the left of

them. The 'Y' indicates that the parameter is enabled and a 'N' indicates that the parameter is disabled. These parameters are changeable except for type '0', which does not allow you to change 'Convert lower to UPPER'. The 'Quote delimiter' parameter is explained in paragraph 3.3.3.5. After the parameters are displayed you will be asked if you wish to change them. If you do, enter 'Y', 'y' or a carriage RETURN. The program will then ask for the setting on all applicable parameters. You will then have the parameters displayed so you can change them again in case you made an error.

### 3.3.3.1 CASE CONVERSION

When designing any Huffman type algorithm it is desirable to keep the number of selections (or branches in the tree) to a minimum. The only difference between an 'a' and an 'A' is simply a matter of case. Both letters convey the same information. Our algorithms (except for type '0') do not include lower case letters. To archive lower case letters in types '1' through '4' we employ a special convention that takes considerably more space in the encoded bit stream. In most programming languages, you cannot use lower case letters except for comments or output strings. Because of these facts we have included the case conversion option.

If the case conversion is enabled, all lower case letters will be automatically converted to upper case and archived in upper case. The only exception to this rule is lower case letters found between quote delimiters as explained in paragraph 3.3.3.5. In our tests we found that compression was increased by as much as 20% by using case conversion thus we recommend using it. If for some reason you don't want to use it, we recommend that you archive your file using type '0'. Remember that with case conversion on, lower case letters are archived as upper case letters and will be restored as upper case letters, not lower case letters.

### 3.3.3.2 END OF FILE (EOF)

The CP/M operating system uses a special character to signify that it has reached the End Of File. This character is the Control - Z character. No, you don't see it on your keyboard but you can find a key on the left labeled 'CTRL'. By holding down that key and depressing the 'Z' key you will generate a value of 26, or Control - Z, to the computer.

This is what is called a non-printable character because it has no corresponding character in the alphabet. By using the Control - Z (sometimes called CTRL-Z) for the End Of File (EOF) marker, CP/M knows that the rest of the data in your file is not used. Since the CP/M system blocks the files in 128 word records, it is possible to have up to 127 words of your file not used. To avoid compressing the unused area of your file, we use the CTRL-Z to flag EOF also.

When the archived file is restored we fill the remaining area at the end of the text with CTRL-Z's automatically. We

recommend that you leave this option enabled at all times unless you are archiving a file that contains CTRL-Z's such as an assembled file (COM type).

#### 3.3.3.3 PASSWORDS

The password function gives you the ability to protect your text from unauthorized access without your permission. In many instances you may have files such as tax records that you don't want other users of your computer system to have access to. If this is the case for you now, you probably remove the disk with the data to a secure place. This is still marginal security. If the disk becomes available to others, then the security is lost. With the password feature of the DENSE-PAC Utility program you do not have to hide your disks or worry about unauthorized access.

When you are asked if you wish to change the archive parameters, if you enable password protection you will be asked for a password to archive and to restore your file. The password may be one to eight (1 - 8) digits long and can be any character on the keyboard that sends a character(s) to the computer. These characters not only include the printable characters but also include the special function keys, ESC and CTRL sequences. When you enter your password, each character that you enter will be echoed back to the screen on the display as an 'X'. If you enter a key that sends two characters to the computer such as a special function key, 'XX' will be displayed.

REMEMBER your password! The restore function will NOT restore a file if you enter your password incorrectly! The password must be entered exactly as you entered it when you archived the file, including the proper case (upper or lower) and anything else. If you were to enter a backspace while entering the original password, it would have to be entered when restoring the file. We suggest that you do not write down or tell anyone your password. Try to use one that you can remember. Don't use simple things such as your phone # or birthday.

If you do not ask for a password in the parameter selection, you will not be asked for one when you archive the file or when you restore the file.

#### 3.3.3.4 KEYWORDS

Every language has some sequence of letters that are commonly repeated in words. In computer languages these are called 'keywords' or 'reserved words'. In our algorithms we have included a searching function that will scan ahead in the text looking for the keywords. You may enable or disable this function but it is recommended that you leave it enabled. When it is enabled the algorithm can compress several letters into one special sequence, thus increasing the compression of your text. The only disadvantage of this function is that all type

'1' through '4' searches are done for upper case letters only. This means that if case conversion is disabled (as explained in paragraph 3.3.3.1), keywords will only match with upper case characters. All type '1' through '4' keywords are restored in upper case only. Type '0' keywords are for lower case only. Keyword search is automatically disabled within a quote sequence as explained in paragraph 3.3.3.5. Refer to appendix A for a list of all keywords used.

### 3.3.3.5 QUOTES

To get off the subject, the case conversion function was included to make the algorithm more efficient as explained in paragraph 3.3.3.1. When it is enabled, it is intended to convert the case on comments that may be included within the program source code. This function works quite well unless there is an encoded output string within the program that is also lower case. Changing the case on the letters within this string would create an undesirable side effect. For example, if you were to archive a program with the following source line in a Basic program :

```
000105 PRINT "This is a output statement"
```

The resultant archived and restored line would be :

```
000105 PRINT "THIS IS A OUTPUT STATEMENT"
```

As you can see, when the program is run, the instruction PRINT will output a line of upper case letters which was not what was intended when the program was written. To get around this problem, the archive function will automatically scan the line for two quote delimiters. Any text between the quote delimiters will be archived as is, that is no case conversion will take place. The only restriction is that the leading and trailing quote delimiters must be on the same line. Since you may have need to use a quote character other than " or ', you may select any character to use for the quote searching function. The preset quotes are listed below for each type.

Type #	Description	Quote
0	= Upper and lower case text	'
1	= Upper case text	'
2	= Assembler source code	'
3	= Pascal source code	'
4	= Basic source code	"

### 3.3.4 INPUT / OUTPUT FILES

Input / output files are standard CP/M file types in the form 'DEV:FNAME.EXT'. As an aid to the user the DEV and/or the EXT will be automatically added to the file name if they are not entered. Listed below are the preset DEV and EXT's.

Type #	Description	DEV	EXT
0	= Upper and lower case text	A	DOC
1	= Upper case text	A	DOC
2	= Assembler source code	A	ASM
3	= Pascal source code	A	PAS
4	= Basic source code	A	BAS
Archive	= Archived output file	A	ARK

If the file does not have an EXT you must enter the period after the file name and the EXT field will be left blank. Below are some examples of entries and their output.

#### INPUT FILES

User entry	Type	Archive output
B:FNAME	0	B:FNAME.DOC
FNAME	2	A:FNAME.ASM
C:NOEXT.	4	C:NOEXT.
D:PASCAL.DOC	1	D:PASCAL.DOC

#### OUTPUT FILES

User entry	Type	Archive output
MYFILE	n/a	A:MYFILE.ARK
YOURFILE.DOC	n/a	A:YOURFILE.DOC

As you can see from the above examples, you can override the preset DEV and / or EXT at the time of entry. The presets are included only as a convenience and do not have to be followed.

### 3.3.5 STATISTICS

The statistics outputted after the archive function is done include the following. All statistical values outputted by the DENSE-PAC Utility are in decimal.

```

Bytes archived           = xxxxx
Keywords archived        = xxxxx
Archived bytes outputted = xxxxx

```

The 'Bytes archived' value is the number of text characters (bytes) read from the input file and archived. If you have a file that is 1 Kbytes long (1024 bytes), this number can vary from 0 to 1024 depending on the number of characters in the file and whether or not you have CTRL-Z code suppression enabled.

The 'Keywords archived' value is the number of keywords found in your text and compressed as a keyword rather than a string of separate characters.

The 'Archived bytes outputted' value is the actual number of 8 bit bytes that the input text was compressed into. Taking this value and the bytes archived value you can use the formula on the next page to determine the amount of compression.



```

Bytes archived           = BIN      ( 1809 )
Archived bytes outputed  = BOUT     ( 979 )
Compression percent      = C

```

$$C = ( 1.0 - ( BOUT / BIN ) ) * 100$$

For example, taking the values in ( ) above and plugging them into the formula you get :

```

C = ( 1.0 - ( 979 / 1809 ) ) * 100
C = ( 1.0 - .541182 ) * 100
C = 45.882 %

```

Remember that this is the compression percent for the outputed bytes versus the inputed bytes. CP/M requires all files to have a multiple records for storage. Remaining portions of records at the end of the files are filled with CTRL-Z codes. In the above example the input file is 2 Kbytes long and the output is 1K. Applying these figures the actual disk space compression is 50.0 %. As the input file increases in size, the difference between the byte compression and the disk compression reduces.

### 3.3.6 AN ARCHIVE TRIP

Here we will take you through a example of the archive function using one of the files on your distribution disk. The file name is DOCFILE.DOC and is an example of various types of text and keywords. To simplify things we will display all your inputs in upper case although you can make any entry in upper or lower case, as you desire. All entries requiring upper case are converted automatically for you. The comments within the dashed lines are for clarification only and are not outputed by the DENSE-PAC Utility or inputed by the user.

```

DENSE-PAC Utility V1.0 (1-Mar-83)
Copyright 1983 by D - Soft
Produced by NEWLINE Software
All Rights Reserved
Enter the function desired :

```

```

A      = Archive file
V      = Archive and verify file
F      = Find best archive algorithm
R      = Restore file
S      = Swap disks
E      = Exit program

```

```

>>> A      | you entered A for archive |

```

Enter the number matching the description of the file you wish to archive :

```

0      = Upper and lower case text
1      = Upper case text
2      = Assembler source code
3      = Pascal source code
4      = Basic source code

```

```

>>> 1      | you entered 1 |

```

Current parameters :

Buffer size	>> XXX	will vary with memory size
Convert to UPPER	>> Y	preset values
Password protection	>> N	dependent
Normal End Of File	>> Y	on the
Keyword searching	>> Y	type you
Quote delimiter	>> '	inputed

Do you wish to change the parameters for archive <Y/N> ? N

You entered N to not change the preset parameters, if you enter a Y, y or a RETURN, you will be asked a series of questions allowing you to change the presets. Once you have answered all of the questions you will be asked again if you wish to change them, giving you the opportunity to correct any errors. The changes you enter only remain in effect during this archive and any subsequent archive using the same type. Each type has a different set of presets.

Enter the file name to archive in the form <A:FNAME.EXT>

>>> x:DOCFILE (RETURN)

You enter x:DOCFILE where 'x' is the disk drive that DOCFILE resides on. DOCFILE is on the distribution disk. Remember that if DOCFILE is on 'A', the device 'A' is not needed nor is the extension 'DOC' since these values are presets.

Enter the file name for the archived output in the form <A.FNAME>

>>> D (RETURN)

You enter D which will generate file A:D.ARK. If you wish a different drive or extension you must enter them also. The file name is anything you want.

If you had enabled password protection, this is the time when you would be asked for your password to include in the archived file.

Working on your files now

-----  
When this message is printed on the display, the archive program is now working. It may take a while if you have a large file so don't panic. It's working hard. While we are waiting we will bring to your notice the 25th line on the bottom of the display. On this line the file status will be displayed. Actually this information showed up after you entered the last file name. It consists of the input and output file names as they are presented to CP/M and the file type. The file type in this case refers to the description that you entered at the beginning of the archive function.  
-----

Archive complete :  
Bytes archived = 6499  
Keywords archived = 380  
Archived bytes outputed = 3746

-----  
When the above statistics are outputed the archive function is complete. The above values are the ones you should receive if you run this example. You are now finished with your trip through archive. You will now get the main menu displayed again.  
-----

The following is a abbreviated version of your trip through archive without out comments.

DENSE-PAC Utility V1.0 (1-Mar-83)  
Copyright 1983 by D - Soft  
Produced by NEWLINE Software  
All Rights Reserved  
Enter the function desired :

A = Archive file  
V = Archive and verify file  
F = Find best archive algorithm  
R = Restore file  
S = Swap disks  
E = Exit program  
>>> A

Enter the number matching the description of the file you wish to archive :

0 = Upper and lower case text  
1 = Upper case text  
2 = Assembler source code  
3 = Pascal source code  
4 = Basic source code  
>>> 1

Current parameters :

```
Buffer size           >> XXX
Convert to UPPER      >> Y
Password protection   >> N
Normal End Of File    >> Y
Keyword searching     >> Y
Quote delimiter       >> '
```

Do you wish to change the parameters for archive <Y/N> ? N  
Enter the file name to archive in the form <A:FNAME.EXT>

>>> x:DOCFILE (RETURN)

Enter the file name for the archived output in the form  
<A.FNAME>

>>> D (RETURN)

Working on your files now

Archive complete :

```
Bytes archived      = 6499
Keywords archived   = 380
Archived bytes outputed = 3746
```

### 3.4 FIND FUNCTION

#### 3.4.1 INTRODUCTION

The find function is used to find the best algorithm to use when archiving your file. The find function will ask you some questions about your file and then compress it with all five algorithms. At the end of the find function, the statistics will be outputed to the screen. From these statistics you will be able to pick the algorithm that best fits your file. This function will NOT archive the file, it only finds the best algorithm. The find function is included as a convenience for you but it does take up to five times longer than the archive function. To select the find function, enter a 'F' as your main menu selection.

#### 3.4.2 FIND PARAMETERS

The parameters that the find function uses are the type '1' presets. You may modify them as explained below and they will be used for each algorithm except for case conversion which is not used for the type '0' algorithm. These parameters are very important for you because they describe your text for the find function. The parameters are listed below:

Buffer size	Convert lower to UPPER
Normal End Of File	Keyword searching
Quote delimiter	

When the parameters are displayed, the current settings will be displayed also. The buffer size is informational only and cannot be changed. This size is the amount of memory available for one buffer (remember that there are 2 buffers). The next 4 parameters will have a 'Y' or a 'N' displayed to the left of them. The 'Y' indicates that the parameter is enabled

and a 'N' indicates that the parameter is disabled. The 'Quote delimiter' parameter is explained in paragraph 3.4.2.4. After the parameters are displayed you will be asked if you wish to change them. If you do, enter 'Y', 'y' or a carriage RETURN. The program will then ask for the setting on all applicable parameters. You will then have the parameters displayed so you can change them again in case you made an error.

#### 3.4.2.1 CASE CONVERSION

If the case conversion is enabled, all lower case letters will be automatically converted to upper case and compressed in upper case. The only exceptions to this rule are lower case letters found between quote delimiters and compression with the type '0' algorithm. Remember that with case conversion on, lower case letters are compressed as upper case letters with type '1' through '4' algorithms.

#### 3.4.2.2 END OF FILE (EOF)

End Of File codes (CTRL-Z) are used in most text files to flag the end of the text. Normally these values will never be in your text. The reason they are normally suppressed is that CP/M fills remaining space in all files with CTRL-Z's. We recommend that you leave this option enabled at all times unless you are compressing a file that contains CTRL-Z codes such as an assembled file (COM type).

#### 3.4.2.3 KEYWORDS

You may enable or disable this function but it is recommended that you leave it enabled. When it is enabled the algorithm can compress several letters into one special sequence thus increasing the compression of your text. The only disadvantage of this function is that all type '1' through '4' searches are done for upper case letters only. This means that if case conversion is disabled (as explained in paragraph 3.4.2.1), keywords will only match with upper case characters. Keyword search is automatically disabled within a quote sequence as explained in paragraph 3.4.2.4. Refer to appendix A for a list of all keywords used.

#### 3.4.2.4 QUOTE STRINGS

Any text between the quote delimiters will be compressed as is, that is no case conversion or keyword searching will take place. The only restriction is that the leading and trailing quote must be on the same line. Since you may have need to use a quote character other than ', you may select any character to use for the quote searching function. The preset quote for type '1' is a " ' ".

#### 3.4.3 INPUT FILE

The input file name is the standard CP/M file type in the form 'DEV:FNAME.EXT'. As an aid to the user the DEV and/or the EXT will be automatically added to the file name if they are not entered. The preset DEV is 'A' and the preset EXT is DOC for the find function.

## 3.4.4 STATISTICS

The statistics outputted when the find function is complete include the following. All statistical values outputted by the DENSE-PAC Utility are in decimal.

Type	0	1	2	3	4
Bytes input	= xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
Keywords	= xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
Bytes output	= xxxxx	xxxxx	xxxxx	xxxxx	xxxxx

The 'Bytes input' value is the number of text characters (bytes) read from the input file and compressed.

The 'Keywords' value is the number of keywords found in your text and compressed as a keyword rather than a string of separate characters.

The 'Bytes output' value is the actual number of 8 bit bytes that the input text was compressed into. This statistic is essentially the bottom line in that the lowest number is the type you should archive the file under. Remember, as explained in the paragraph 3.3.5, that this is not the number of records needed for the storage of the file if it were archived. In reality the number of records for all 5 types may be the same.

## 3.4.5 AN FIND TRIP

Here we will take you through a example of the find function using one of the files on your distribution disk. The file name is DOCFIE.DOC and is an example of various types of text and keywords. To simplify things we will display all your inputs in upper case although you can make any entry in upper or lower case as you desire. All entries requiring upper case are converted automatically for you. The comments within the dashed lines are for clarification only and are not outputted by the DENSE-PAC Utility or inputted by the user.

DENSE-PAC Utility V1.0 (1-Mar-83)

Copyright 1983 by D - Soft

Produced by NEWLINE Software

All Rights Reserved

Enter the function desired :

```

A      = Archive file
V      = Archive and verify file
F      = Find best archive algorithm
R      = Restore file
S      = Swap disks
E      = Exit program

```

>>> F

```

-----
| you entered F for find |
-----

```

Current parameters :

```

Buffer size      >> XXX | will vary with memory size
Convert to UPPER  >> Y  | preset values
Normal End Of File >> Y  | are for
Keyword searching >> Y  | type '1'
Quote delimiter  >> '  |
-----

```

Do you wish to change the parameters for find <Y/N> ? Y

-----  
You entered Y to change the preset parameters, if you had entered a N or n, you would not be asked the questions below. We are entering Y to change the preset to convert to UPPER. Notice below that you cannot select password protection since this option is disabled during the find function. The changes you enter only remain in effect during this find. Note also that you can leave the preset to convert to UPPER but this is disabled for type '0' automatically.  
-----

Do you wish lower case letters converted to UPPER case <Y/N>? N

-----  
You entered N to disable conversion.

Is CTRL-Z used for End Of File (EOF) < Y/N > ? Y

-----  
You entered Y to enable using EOF.

Do you wish keyword searching < Y/N > ? Y

-----  
You entered Y for keyword searching

Do you wish to change the QUOTE delimiter < Y/N > ? N

-----  
You entered N to leave the quote delimiter as it is. If you enter a Y, y or a carriage RETURN, you would be asked to enter the new quote delimiter. The new quote delimiter is limited only to any single ASCII character that you can enter from the keyboard.  
-----

Current parameters :

Buffer size	>> XXX	----- You will notice that the convert to UPPER preset has now been changed to N. These parameters will be used for find on types '0' through '4'. -----
Convert to UPPER	>> N	
Normal End Of File	>> Y	
Keyword searching	>> Y	
Quote delimiter	>> '	

Do you wish to change the parameters for find <Y/N> ? N

-----  
 You entered N to leave the preset parameters as they are now. If you had made an error (heaven forbid), you can again change them and can do so until you are satisfied.  
 -----

Enter the file name to find the best algorithm for in the form  
 <A:FNAME.EXT>

>>> x:DOCFILE (RETURN)

-----  
 You enter x:DOCFILE where 'x' is the disk drive that DOCFILE resides on. DOCFILE is on the distribution disk. Remember that if DOCFILE is on 'A', the device 'A' is not needed nor is the extension 'DOC' since these values are presets.  
 -----

Working on your files now  
 Working on your files now  
 Working on your files now  
 Working on your files now  
 Working on your files now

-----  
 No our word processor is not crazy! Each one of the above duplicate messages is outputted when the find function is compressing your file with a different type. While we are waiting we will bring to your notice the 25th line on the bottom of the display. On this line the file status will be displayed. Actually this information showed up after you entered the file name. It consists of the input file as it is presented to CP/M and the output file is 'None' since the find function does not output a file. The type is updated every time a new type is tried and serves as a indication of how far the find function has gone.  
 -----

Archive complete :

Type	=	0	1	2	3	4
Bytes input	=	6499	6499	6499	6499	6499
Keywords	=	260	112	70	90	100
Bytes output	=	3960	6202	6211	6064	6006

-----  
 When the above statistics are outputted the find function is completed. These values are similar to what you should receive if you run this example. You will notice (we hope) the significant advantage of using type '0' to archive this file with the presets we selected. You will find that if you run this example with the original presets (lower to UPPER enabled) that type '3' will be the best choice. You are now finished with your trip through find. You will now get the main menu displayed again.  
 -----



The following is an abbreviated example of your trip through find without our comments.

DENSE-PAC Utility V1.0 (1-Mar-83)

Copyright 1983 by D - Soft

Produced by NEWLINE Software

All Rights Reserved

Enter the function desired :

A = Archive file  
 V = Archive and verify file  
 F = Find best archive algorithm  
 R = Restore file  
 S = Swap disks  
 E = Exit program

>>> F

Current parameters :

Buffer size >> XXX  
 Convert to UPPER >> Y  
 Normal End Of File >> Y  
 Keyword searching >> Y  
 Quote delimiter >> '

Do you wish to change the parameters for find <Y/N> ? Y

Do you wish lower case letters converted to UPPER case <Y/N>? N

Is CTRL-Z used for End Of File (EOF) < Y/N > ? Y

Do you wish keyword searching < Y/N > ? Y

Do you wish to change the QUOTE delimiter < Y/N > ? N

Current parameters :

Buffer size >> XXX  
 Convert to UPPER >> N  
 Normal End Of File >> Y  
 Keyword searching >> Y  
 Quote delimiter >> '

Do you wish to change the parameters for find <Y/N> ? N

Enter the file name to find the best algorithm for in the form

<A:FNAME.EXT>

>>> x:DOCFILE (RETURN)

Working on your files now

Working on your files now

Working on your files now

Working on your files now

Working on your files now

Archive complete :

Type	=	0	1	2	3	4
Bytes input	=	6499	6499	6499	6499	6499
Keywords	=	260	112	70	90	100
Bytes output	=	3960	6202	6211	6064	6006

### 3.5 RESTORE FUNCTION

#### 3.5.1 INTRODUCTION

The restore function is used to restore your text to its original format. The original format being the format within the bounds of the parameters of the archived file. If case conversion was selected when you archived the file, all of the original lower case letters not within quote delimiters would be upper case. To select the restore function, enter a 'R' as your main menu selection.

#### 3.5.2 INPUT / OUTPUT FILES

Input / output files are standard CP/M file types in the form 'DEV:FNAME.EXT'. As an aid to the user the DEV and / or the EXT will be automatically added to the file name if they are not entered. When you originally archived the file you are restoring, the file type was archived with the file. The restore function recovers this type and uses it for the selection of defaults. Listed below are the preset DEV and EXT's.

Type #	Description	DEV	EXT
Input file			
Archive	= Archived input file	A	ARK
Output file			
0	= Upper and lower case text	A	DOC
1	= Upper case text	A	DOC
2	= Assembler source code	A	ASM
3	= Pascal source code	A	PAS
4	= Basic source code	A	BAS

If the file does not have an EXT you must enter the period after the file name and the EXT field will be left blank. Below are some examples of entries and their output.

Input file name entered

User entry	Type	Restore output to CP/M
B:FNAME	n/a	B:FNAME.ARK
FNAME	n/a	A:FNAME.ARK
C:NOEXT.	n/a	C:NOEXT.
D:PASCAL.DOC	n/a	D:PASCAL.DOC

Output file name entered

User entry	Type*	Restore output to CP/M
MYFILE	2	A:MYFILE.ASM
YOURFILE.DOC	4	A:YOURFILE.DOC

\* The type is extracted from the file to be restored

As you can see from the above examples, you can override the preset DEV and / or EXT at the time of entry. The presets are included only as a convenience and do not have to be followed.

### 3.5.3 STATISTICS

The statistics outputted after the restore function is completed include the following. All statistical values outputted by the DENSE-PAC Utility are in decimal.

Bytes inputted	=	xxxxxx
Keywords found	=	xxxxxx
Restored bytes outputted	=	xxxxxx

The 'Bytes inputted' value is the number of bytes read from the input file and restored. This value should be the same as 'Archived bytes outputted' when you archived the file.

The 'keywords found' value is the number of keywords found in the input file that were compressed as a keyword. This value should also reflect the 'Keywords archived' statistic reported to you when the file was archived.

The 'Restored bytes outputted' value is the actual number of 8 bit bytes that were outputted to the restored file. This figure does not include any trailing CTRL-Z codes to pad out the output file.

### 3.5.4 A RESTORE TRIP

Here we will take you through a example of the restore function using one of the files on your distribution disk. The file name is DOC.ARK and is an example of various types of text and keywords. To simplify things we will display all your inputs in upper case although you can make any entry in upper or lower case as you desire. All entries requiring upper case are converted automatically for you. The comments within the dashed lines are for clarification only and are not outputted by the DENSE-PAC Utility or inputted by the user.

DENSE-PAC Utility V1.0 (1-Mar-83)

Copyright 1983 by D - Soft

Produced by NEWLINE Software

All Rights Reserved

Enter the function desired :

A	= Archive file
V	= Archive and verify file
F	= Find best archive algorithm
R	= Restore file
S	= Swap disks
E	= Exit program

>>> R

-----  
you entered R for restore

Output resulting text to the screen < Y/N > ? Y

-----  
You enter a 'Y' for YES you want to see the text as it is restored. The text is also outputted to the file you will name in a step or two, so don't worry that it won't get saved. This option is not mandatory and does take some extra processing time due to the speed of the display. If you entered 'N' or 'n' for NO, you would not be asked the next two questions.  
-----

Continuous scrolling of the output < Y/N > ? N

-----  
You entered a 'N' for NO you don't want to watch the text go flying by. In this mode, the restore function will output 22 lines of restored text and then ask you to 'Enter any key to continue >>>'. This gives you a chance to look at the text before it scrolls off the screen. Even if the display is slow by the computers standard, it sure goes faster than we can read! If you had entered a 'Y', 'y' or carriage RETURN, the text is continuously outputted to the screen until done. While outputting text the restore function may quit in the middle of a line. Don't panic! The program has simply run out of text in the buffer and has to restore another buffer before it can continue output to the screen.  
-----

Do you want the output forced to 80 columns < Y/N > ? N

-----  
You entered a 'N' for NO you don't want the text forced to 80 columns. Normally your text will have line feeds before it reaches column 80 on the screen so this option does not really get you anything. You probably prefer to see the text as it was archived. The display is set up to 255 columns so if your text is past 80 columns, it will not be displayed. If this is the case, you can enter 'Y', 'y' or carriage RETURN to see all of the text. When the restore function is forcing 80 column text, it will count characters on a line and insert a CR / LF sequence in the display output only. The outputted file will not have the CR / LF sequence inserted in it.

The file DOC.ARK does have a line extending past 80 columns so we recommend that you take this trip again but enter a 'Y' for this option to see the difference.  
-----

Enter the file name to restore in the form <A:FNAME.EXT>

>>> x:DOC (RETURN)

-----  
You enter x:DOC where 'x' is the disk drive that file DOC resides on. File DOC is on the distribution disk. Remember that if file DOC is on 'A', the device 'A' is not needed nor is the extension 'ARK' since these values are presets.  
-----

Enter the file name for the restored output in the form

<A.FNAME>

>>> D (RETURN)

-----  
You enter D which will generate file A:D.DOC. If you wish a different drive or extension you must enter them also.  
-----

-----  
If you had enabled password protection, this is the time when you would be asked for your password. No password is included in the distribution file. If there was a password and you entered it incorrectly, the restore function will abort execution and return to the main menu.  
-----

Working on your files now

-----  
When this message is printed on the display, the restore function is now working. It may take a while if you have a large file so don't panic. It's working hard. If you have screen output enabled, the text will start as soon as one complete output buffer is processed. If your archived file included any non-printable or non-ASCII characters, these will be outputted as a graphic 'X'. If scrolling is enabled you will be asked to 'Enter any key to continue >>>' after 22 lines have been outputted. This gives you time to see the text before it is scrolled off the screen.  
-----

Restore complete :

Bytes inputted = 3746

Keywords found = 380

Restored bytes outputted = 6499

-----  
When the above statistics are outputted the restore function is complete. The above values are the ones you should receive if you run this example. You are now finished with your trip through restore. You will now get the main menu displayed again.  
-----

The following is an abbreviated trip through restore without our comments.

DENSE-PAC Utility V1.0 (1-Mar-83)

Copyright 1983 by D - Soft

Produced by NEWLINE Software

All Rights Reserved

Enter the function desired :

A = Archive file

V = Archive and verify file

F = Find best archive algorithm

R = Restore file  
S = Swap disks  
E = Exit program

>>> R

Output resulting text to the screen < Y/N > ? Y

Continuous scrolling of the output < Y/N > ? N

Do you want the output forced to 80 columns < Y/N > ? N

Enter the file name to restore in the form <A:FNAME.EXT>

>>> x:DOC (RETURN)

Enter the file name for the restored output in the form  
<A.FNAME>

>>> D (RETURN)

Working on your files now

( your text is displayed here )

Restore complete :

Bytes inputed = 3746

Keywords found = 380

Restored bytes outputed = 6499

### 3.6 ARCHIVE / VERIFY FUNCTION

#### 3.6.1 INTRODUCTION

The archive / verify function is used to compress your text files and write out the compressed version on your disk. When the archive portion is completed, the verify portion will read in the archived file and compare it to the original file. This allows you to verify that the archived file is a true copy and that it can be restored at a later date. The archive portion of this function is identical to the archive function which is fully explained in paragraph 3.3. The verify portion is a subset of the restore function and will be explained in the following paragraphs.

The file created by the archive / verify function is identical to the one created by the archive function. The advantage of using this function over the archive function is that the restoring of the file is tested immediately. The disadvantage of the archive / verify function is that it increases the time necessary to archive a file.

What the verify portion actually does is to read in the archived file, restore it using the restore function and then compare byte by byte the restored information to the original information.

Once the archive portion of archive / verify is completed, you will be asked information about displaying the results of the verify portion. The information requested is almost identical to restore. The only difference is that you will be asked one additional question if you select not to have the results of the verify outputed to the screen. You will be asked 'Do you want to stop if a byte error is found'. This is necessary because you have asked not to be shown the results and the DENSE-PAC Utility needs to know if you want byte errors to be shown or scrolled off the screen. In either case the

byte error will be displayed. To select the archive / verify function, enter a 'V' as your main menu selection.

### 3.6.2 VERIFY OUTPUTS

Although the requesting of display parameters is similar in restore and in verify, the outputs will differ. Two differences may occur while verifying your file. .

The first arises when you have selected lower case to UPPER case character conversion. If you have, there will be a case difference between the archived file and the original file. When there is a case difference, the output character will be displayed on your screen in reverse video. The characters displayed are always from the original file. If you see a lot of reverse video lower case letters, it is caused by the case conversion. This is not an error so don't worry.

The second difference you may see is a byte error report. While comparing the text files, if two characters are not identical or case conversions, you will receive a message on the screen :

```
'Byte VVVVV incorrect, --- Original byte = WWW > X <
restored byte = YYY > Z <'
```

Where

VVVVV = The byte number in error.  
WWW = The decimal value of the original byte.  
X = The ASCII equivalent of WWW or a reverse video 'X' if the value is not printable.  
YYY = The decimal value of the restored byte.  
Z = The ASCII equivalent of YYY or a reverse video 'X' if the value is not printable.

Normally you will never receive this error, but if you do it means that the file is not restoreable and should be deleted. The best way to handle the error is to enter a Control - B to abort the operation and get back to the menu. Once there you can try the operation again.

The most common cause of this error is when you archive a file with embedded CTRL-Z codes such as machine code files. If the suppression of CTRL-Z codes was enabled (which is the preset), the CTRL-Z codes will be missing from the archived file and a mismatch will occur giving a byte error. This is easily observed by looking at the byte errors reported. The first one will probably show that the original file byte was 26 which will be missing from the restored file. If this is the case, abort the operation and re-archive the file with CTRL-Z suppression disabled. CTRL-Z codes are the only byte values that can be suppressed as explained in paragraph 3.3.3.2, all other byte values (0 to 255) can be archived.

### 3.6.2 VERIFY STATISTICS

Two sets of statistics will be output by the archive / restore function. The first set is the result of the archive and is identical to the normal archive statistics as explained in paragraph 3.3.5. The second set of statistics occur after the verify function is completed. All statistical values outputted by the DENSE-PAC Utility are in decimal. Below is the format of these statistical outputs.

Verification complete :

Bytes archived	= xxxxx
Bytes verified	= xxxxx
Lower to UPPER case conversions	= xxxxx
Byte errors	= xxxxx

The 'Bytes archived' value is the number of bytes read from the input file and archived.

The 'Bytes verified' is the number of bytes that the file was restored to. Both the bytes archived and the bytes verified should be the same value.

The 'Lower to UPPER case conversions' is the number of case mis-matches that we found. This is not an error! It's presented as information only.

The 'Byte errors' is the number of errors found while doing the verify portion. If this value is not zero, the archived file should be deleted as it cannot be properly restored.

### 3.6.4 AN ARCHIVE / VERIFY TRIP

Here we will take you through a example of the archive / verify function using one of the files on your distribution disk. The file name is DOCFIELD.DOC and is an example of various types of text and keywords. To simplify things we will display all your inputs in upper case although you can make any entry in upper or lower case as you desire. All entries requiring upper case are converted automatically for you. The comments within the dashed lines are for clarification only and are not outputted by the DENSE-PAC Utility or inputted by the user.

DENSE-PAC Utility V1.0 (1-Mar-83)

Copyright 1983 by D - Soft

Produced by NEWLINE Software

All Rights Reserved

Enter the function desired :

A	= Archive file
V	= Archive and verify file
F	= Find best archive algorithm
R	= Restore file
S	= Swap disks
E	= Exit program

>>> V

-----  
you entered V for archive / verify



Enter the number matching the description of the file you wish to archive :

0 = Upper and lower case text  
 1 = Upper case text  
 2 = Assembler source code  
 3 = Pascal source code  
 4 = Basic source code

>>> 2

-----  
you entered 2

Current parameters :

Buffer size	>> XXX	will vary with memory size
Convert to UPPER	>> Y	preset values
Password protection	>> N	dependent
Normal End Of File	>> Y	on the
Keyword searching	>> Y	type you
Quote delimiter	>> '	inputed

-----

Do you wish to change the parameters for archive <Y/N> ? Y

-----  
 | You entered Y to change the preset parameters, if you had  
 | entered a N or n , you would not be asked the questions  
 | below. We are entering Y to change the preset to include  
a password.

Do you wish lower case letters converted to UPPER case <Y/N>? Y

-----  
You entered Y to enable conversion.

Do you wish password protection < Y/N > ? Y

-----  
You entered Y to enable passwords protection

Is CTRL-Z used for End Of File (EOF) < Y/N > ? Y

-----  
You entered Y to for normal EOF.

Do you wish keyword searching < Y/N > ? Y

-----  
You entered Y for keyword searching

Do you wish to change the QUOTE delimiter < Y/N > ? N

-----  
You entered N to leave the quote delimiter as it is. If you enter a Y, y or a carriage RETURN, you would be asked to enter the new quote delimiter. The new quote delimiter is limited only to any single ASCII character that you can enter from the keyboard.  
-----

Current parameters :

Buffer size	>> XXX	-----	You will notice that the
Convert to UPPER	>> Y		password protection is now
Password protection	>> Y		changed to Y.
Suppress NULL (Ø) codes	>> Y	-----	
Keyword searching	>> Y		
Quote delimiter	>> '		

Do you wish to change the parameters for archive <Y/N> ? N

-----  
You entered N to leave the preset parameters as they are now. If you had made an error (heaven forbid), you can again change them and can do so until you are satisfied.  
-----

Enter the file name to archive in the form <A:FNAME.EXT>

>>> x:DOCFILE.DOC (RETURN)

-----  
You enter x:DOCFILE where 'x' is the disk drive that DOCFILE resides on. DOCFILE is on the distribution disk. Remember that if DOCFILE is on 'A', the device 'A' is not needed. The extension 'DOC' is needed since the preset extension is ASM.  
-----

Enter the file name for the archived output in the form

<A.FNAME>

>>> V (RETURN)

-----  
You enter V which will generate file A:V.ARK. If you wish a different drive or extension you must enter them also. The file name is anything you want.  
-----

Please enter your password >>> XXXXXX (HEATH and RETURN)

-----  
No you did not enter 6 X's. The password you entered was HEATH and a carriage RETURN. The password function echoed your 6 inputs as X's. Passwords are never displayed.  
-----

Working on your files now

-----  
When this message is printed on the display, the archive  
program is now working.  
-----

Archive complete :  
Bytes archived = 6499  
Keywords archived = 145  
Archived bytes outputed = 3846

-----  
When the above statistics are outputed the archive  
function is complete. The above values are the ones you  
should receive if you run this example. You are now  
finished with the archive portion of the archive / verify  
function. The verify portion is next.  
-----

Archiving complete - verifying file

Output resulting text to the screen < Y/N > ? Y

-----  
You enter a 'Y' for YES you want to see the text as it is  
verified. This option is not mandatory and does take some  
extra processing time due to the speed of the display. If  
you entered 'N' or 'n' for NO, you would not be asked the  
next two questions.  
-----

Continuous scrolling of the output < Y/N > ? N

-----  
You entered a 'N' for NO you don't want to watch the text  
go flying by. In this mode, the verify function will  
output 22 lines of verified text and then ask you to  
'Enter any key to continue >>>'. This gives you a chance  
to look at the text before it scrolls off the screen. Even  
if the display is slow by the computers standard, it sure  
goes faster than we can read! If you had entered a 'Y',  
'y' or carriage RETURN, the text is continuously outputed  
to the screen until done. While outputing text the  
verify function may quit in the middle of a line. Don't  
panic! The program has simply run out of text in the  
buffer and has to restore another buffer before it can  
continue verifying.  
-----

Do you want the output forced to 80 columns < Y/N > ? Y

You entered a 'Y' for yes you want the output forced to 80 columns. Normally your text will have line feeds before it reaches column 80 on the screen so this option does not really get you any thing, but the file you are verifying does have a line that is over 80 columns long so now you have the chance to see this option work.

The input and output file names do not have to be entered again since this information is saved from the archive portion. This also applies to the password if you have one which is checked but not displayed. We had one.

Working on your files now

When this message is printed on the display, the verify function is now working. It may take a while if you have a large file so don't panic. It's working hard. If you have screen output enabled, the text will start as soon as one complete input buffer is processed. If your archived file included any non-printable or non-ASCII characters, these will be outputted as a graphic 'X'. If scrolling is enabled you will be asked to 'Enter any key to continue >>>' after 22 lines have been outputted. This gives you time to see the text before it is scrolled off the screen. Remember that case conversions will be displayed in lower case reverse video.

Verification complete :  
Bytes archived = 6499  
Bytes verified = 6499  
Lower to UPPER case conversions = 2472  
Byte errors = 0

When the above statistics are outputted the archive / verify function is completed. The above values are the ones you should receive if you run this example. You are now finished with you trip through archive / verify. you will now get the main menu displayed again.

The following is the complete archive / verify trip in a abbreviated form with out the comments.

DENSE-PAC Utility V1.0 (1-Mar-83)  
Copyright 1983 by D - Soft  
Produced by NEWLINE Software  
All Rights Reserved

Enter the function desired :

A = Archive file  
 V = Archive and verify file  
 F = Find best archive algorithm  
 R = Restore file  
 S = Swap disks  
 E = Exit program

>>> V

Enter the number matching the description of the file you wish to archive :

0 = Upper and lower case text  
 1 = Upper case text  
 2 = Assembler source code  
 3 = Pascal source code  
 4 = Basic source code

>>> 2

Current parameters :

Buffer size >> XXX  
 Convert to UPPER >> Y  
 Password protection >> N  
 Normal End Of File >> Y  
 Keyword searching >> Y  
 Quote delimiter >> '  
 Do you wish to change the parameters for archive <Y/N> ? Y  
 Do you wish lower case letters converted to UPPER case <Y/N>? Y  
 Do you wish password protection < Y/N > ? Y  
 Is CTRL-Z used for End Of File (EOF) < Y/N > ? Y  
 Do you wish keyword searching < Y/N > ? Y  
 Do you wish to change the QUOTE delimiter < Y/N > ? N

Current parameters :

Buffer size >> XXX  
 Convert to UPPER >> Y  
 Password protection >> Y  
 Suppress NULL (0) codes >> Y  
 Keyword searching >> Y  
 Quote delimiter >> '

Do you wish to change the parameters for archive <Y/N> ? N

Enter the file name to archive in the form <A:FNAME.EXT>

>>> x:DOCFIELD.DOC (RETURN)

Enter the file name for the archived output in the form <A.FNAME>

>>> V (RETURN)

Please enter your password >>> XXXXXX (HEATH and RETURN)

Working on your files now

Archive complete :

Bytes archived = 6499  
 Keywords archived = 145  
 Archived bytes outputed = 3846

Archiving complete - verifying file

Output resulting text to the screen < Y/N > ? Y  
Continuous scrolling of the output < Y/N > ? N  
Do you want the output forced to 80 columns < Y/N > ? Y  
Working on your files now

( your text is displayed )

Verification complete :

Bytes archived	=	6499
Bytes verified	=	6499
Lower to UPPER case conversions	=	2472
Byte errors	=	0

### 3.7 SWAP FUNCTION

#### 3.7.1 INTRODUCTION

The swap function is used to exchange disks without having to exit to the operating system (CP/M). When you enter 'S' for your main menu entry, the swap function will call CP/M and ask it to reset all of the disk drives on the system. This allows you to change disks without the Read Only error.

When you enter swap, the DENSE-PAC Utility will print out a warning message that all of your disks are now read / write and that you may now swap your disks. Do not try to exchange your disks until told to do so.

### 3.8 OOPS ERRORS

#### 3.8.1 INTRODUCTION

Through out this document errors have been mentioned and what to do about them. In this section we will explain all of the possible errors and what you can do to correct them. Obviously we probably will miss some so we will first give you some general guidance to try to solve the unmentioned errors that can occur.

Errors generally fall into two categories, operator input errors and program detected errors. The operator input errors generally occur when you, as the operator, misunderstand the directions given and make an incorrect entry.

When you make a mistake two things can happen. The first is that the input that you made is not within the limits that the program sets for the particular input so an error message is issued. In all cases when you are required to make an input to the program, your entry will be case checked. Case checking means that the program will check to see that your input falls within the limits for the input. When you make an entry that is not allowed, you will be notified with an error message in reverse video and have the bell ring. You will then be asked to re-enter the input.

The second class of errors are those that the program detects as being illogical conditions. These may be such things as not enough memory to complete a function or bad status from a disk read. All of these errors are also reported in the same way as the operator input errors. System detected errors that return are displayed with a system error message. When any of these errors occur, the best way to recover is to enter a Control - B and abort to the beginning of the program. You may then retry the function you were attempting. If the errors are persistent and deals with system errors such as bad disk reads, you had better get you system looked at.

### 3.8.2 ERRORS EXPLAINED

The following paragraphs list all the possible error messages that the DENSE-PAC Utility can output and brief descriptions of what these errors indicate.

#### 3.8.2.1 OPERATOR ERRORS AND WARNING MESSAGES

'Warning - all disks are read/write'

'You may change disks now'

You have swaped disks which causes all disks to be considered read / write. This does not over ride the write protect tab on the disk.

'Not enough memory for verify - aborting'

You must have at least 1024 addresses of memory between the DENSE-PAC Utility and the bottom of BDOS for buffers.

'The input file is not a archived file!'

You have tried to restore a file that was not originally archived. Re-enter the correct file name when asked to. If it is the correct file name you file is bad.

'Your password is incorrect - aborting'

You have entered the wrong password. You will have to start over again and enter the correct password.

'Cannot open file'

The file name that you entered does not exist on the device you specified. Re-enter the correct file name.

'Byte xxxxx --- Original byte = xxx > x < restored byte = xxx > x <'

This is a verify error and is explained in paragraph 3.6.2.

'x is not a correct choice - try again'

You have made a incorrect choice when asked to input a single character selection. The incorrect input is "x".

### 3.8.2.2 PROGRAM DETECTED HARDWARE ERRORS

All of the following hardware type errors are disk related. If any of them occur we suggest that you refer to the CP/M manual in the case of system error messages. If the error does not seem to be a valid hardware error please do not hesitate to write to us with all the available information and we will try to resolve the matter.

'Archive file is corrupt - aborting'

The structure expected for an archive file was incorrect possibly due to some one trying to crack it. The End of Block code was not found where it should be. You can retry but it probably won't work.

'File error - aborting'

The system detected an error while attempting to read or write a file. The file has been opened so it exists.

### 3.8.2.1 PROGRAM DETECTED SOFTWARE ERRORS

We have tried our hardest to catch all possible bugs in the program but it seems inevitable that bugs do creep in from some where. In an effort to catch all bugs, we have left several error traps in the code to catch all of these errors. If one of the errors happen to you, we would very much appreciate if you would write down all the related information and send in all the information. We will try to recreate the error and if we can recreate it, we will fix the bug.

'The input file is not a archived file!'

This is a program bug error ONLY when it is output during a VERIFY function. It is NOT a program bug when restoring a file.

'Master file failed to open - aborting'

This error occurs during the verify function and indicates that the original file archived fails to open for a read after it has been archived. This could be a hardware error as well as a program bug.



'Unrecoverable restore error # x - aborting'

There are 5 of these as listed below.

Where 'x' =

1,3,5 = All of the archive file has been read and no  
End Of File marker has been found.

2,4,6 = Last read of the archive file did not get any  
data and EOF has not been found.

If any of these occur take down the information and  
write us. We will also need a copy of the archive file  
to find the error.

A friendly warning to those of you that cannot keep away from tinkering with things. If you try to crack the code of an archived file and change ANY of the bits in the file, it will not restore!! Please do not write us asking to restore anything that has been tinkered with. It is not possible.

If you have forgotten your password, we will, for a nominal fee, crack out the password for you. To do this you must submit your registration number and your problem (forgotten password). We will then inform you to send in your disk and the fee. We will not restore the file. The program we use will only crack out the password.

Further information about the program and how it works is not available. Due to the security features in the program the source code will not be released to any one including the distributor. If you have a valid question or any comments about what you like or dislike please write. We will attempt to answer all reasonable questions. If future releases with enhancements are released, we will keep the same algorithms or issue a cross translation program. Don't be timid. Archive all those text files.

## APPENDIX A

Following are the type '0' keywords for lower case text.

ase	and	are	all	ate	cal	can	car
cat	col	com	con	cor	cou	cro	dec
dis	ele	exp	ent	est	ess	ere	for
ful	gen	gra	har	hea	her	inc	ind
ins	int	ion	ive	ill	ist	ing	man
met	mar	mis	mon	men	non	ove	our
par	pen	per	pre	pri	pro	rep	sta
ste	sto	str	sub	sup	tha	thi	the

Following are the type '1' keywords for UPPER case text.

CAT	COL	COM	CON	COR	COU	CRO	DEC
DIS	ELE	EXP	ENT	EST	ESS	ERE	FOR
FUL	GEN	GRA	HAR	HEA	HER	INC	IND
INS	INT	ION	IVE	ILL	IST	ING	MAN
MET	MAR	MIS	MON	MEN	NON	OVE	OUR
PAR	PEN	PER	PRE	PRI	PRO	REP	STA
STE	STO	STR	SUB	SUP	THA	THI	THE

Following are the type '2' keywords for Assembly code.

ACI	ADC	ADD	ADI	ANA	ANI	CALL	CMA
CMC	CMP	CNC	CNZ	CNE	CPE	CPI	CPO
DAD	DCR	DCX	INR	INX	JMP	JNC	JNZ
JNE	JPE	JPO	LDAX	LDA	LXI	LHLD	MVI
MOV	NOP	ORA	ORI	OUT	PCHL	POP	PSW
PUSH	RAL	RAR	RET	RLC	RNC	RNZ	RPE
RPO	RRC	RST	SBB	SBI	SHLD	SPHL	STAX
STA	STC	SUB	SUI	XCHG	XRA	XRI	XTHL

Following are the type '3' keywords for PASCAL code.

AND	ARRAY	ABS	BEGIN	BOOLEAN
CASE	CONST	CHAR	CLOSE	DIV
DOWNT0	DISPOSE	ELSE	END	EOF
EOLN	EXTERN	FILE	FOR	FUNCTION
FALSE	FORWARD	FREE	GOTO	INTEGER
INPUT	LABEL	MOD	MAXINT	NEW
NIL	NOT	ODD	OPEN	ORD
OUTPUT	PROCEDURE	PROGRAM	POS	PRED
RECORD	REPEAT	READLN	READ	REAL
RESET	REWRITE	ROUND	SET	SUCC
STRING	SQR	THEN	TYPE	TRUE
TEXT	TRUNC	UNTIL	VAR	WHILE
WITH	WRITELN	WRITE	XOR	

## APPENDIX A ( continued )

Following are the type '4' keywords for BASIC code.

ABS	ASC	ATN	AND	CHAIN
CHR\$	CLOSE	COS	COMMON	DATA
DEF	DIM	END	ERROR	EXP
ELSE	EOF	ERR	FOR	FRE
GOSUB	GOTO	HEX\$	INPUT	INT
LEFT\$	LEN	LINE	LOG	MID\$
MKD\$	MKI\$	MK\$	MOD	NEXT
NOT	OPEN	OUT	OCT\$	PEEK
POKE	POS	PRINT	READ	REM
RESTORE	RETURN	RIGHT\$	RND	RANDOM
SGN	SIN	SPC	STEP	STOP
STRING\$	TAB	TAN	THEN	VAL
WEND	WHILE	WIDTH	WRITE	