



apple software bank

diskware™

LEMENTARY, MY DEAR APPLE
EDUCATION SERIES
D0015
APPLE 1980

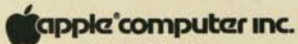


680.0044.01




REQUIRES  SECTOR DISK DRIVES

Some Diskware Packages may contain
only one Diskette.



10260 Bandley Drive,
Cupertino, California 95014

© 1979 Apple Computer Inc.



Some Diskware Packages may contain
only one Diskette.

 **apple computer inc.**

10260 Bandley Drive,
Cupertino, California 95014

© 1979 Apple Computer Inc.



diskware™

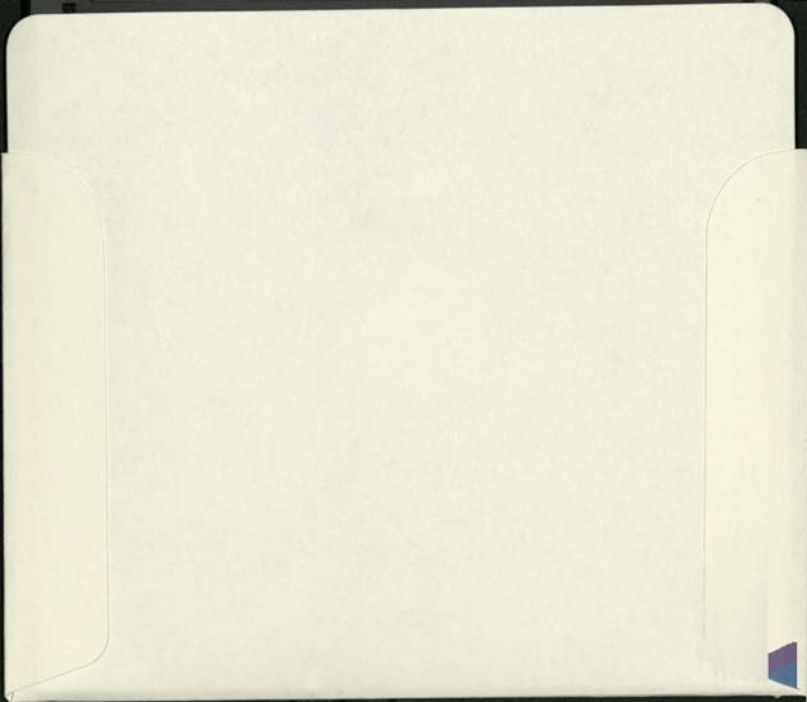
ELEMENTARY, MY DEAR APPLE™
EDUCATION SERIES
A2D0015

© APPLE 1980



680.0044.01

isk II™



Apple II



Elementary, My Dear Apple



NOTICE

Apple Computer Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

APPLE COMPUTER INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. APPLE COMPUTER INC. SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT APPLE COMPUTER INC., ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL APPLE COMPUTER INC. BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF APPLE COMPUTER INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer Inc.

©1980 by APPLE COMPUTER INC.
10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

The word APPLE and the Apple logo are registered trademarks of APPLE COMPUTER INC.

APPLE Product #A2L0015
(030-0114-00)

TABLE OF CONTENTS

INTRODUCTION

THE GAMES

PROGRAMMER'S CORNER

INSIDE DON'T FALL

INDEX

Apple II

Elementary, My Dear Apple

TABLE OF CONTENTS

INTRODUCTION

1

CHAPTER 1

THE GAMES

3

- 3 DARTS
- 7 LEMONADE
- 8 SUPERMATH
- 9 DON'T FALL

CHAPTER 2

PROGRAMMER'S CORNER: INSIDE DON'T FALL

11

- 9 Getting Inside
- 10 Anatomy of All Shell Games
- 11 The DON'T FALL Word List
- 12 Special Word Lists
- 13 The Problem Entry Editor
 - 13 Entering a New Category
 - 16 Editing an Existing Category
 - 17 Back to the Main Program
 - 18 Searching for a Category
 - 19 Verifying the Data Table
 - 19 Replacing Old Categories
 - 20 Inserting and Deleting Lists
- 20 Leaving the Editor
- 21 An Important Fringe Benefit

INDEX

25

INTRODUCTION

The ELEMENTARY, MY DEAR APPLE diskette contains four colorful games for kids from age 9 to 99. These games are both fun and educational. They're fun because they use lots of color animation and sound effects, and they're educational because they deal with manipulating words and numbers.

You don't need to know anything about computers to play the games. All the games will give you instructions to tell you how to play them. They'll guide you step by step, asking you questions, giving you advice.

This manual is divided into sections that tell you what each game is about and how to play it. The discussion won't lead you step by step--the games themselves do that. In addition, sections of DARTS and DON'T FALL tell teachers, parents, and other interested folk how to modify the games to their liking. You can control whether or not DARTS is musical, and you can change the list of words that DON'T FALL uses.

To use these programs, you will need:

- * An Apple II or Apple II Plus computer, with at least 48K of memory.
- * A Disk II controller card and at least one disk drive.
Elementary, My Dear Apple uses sixteen sector DOS, so you'll need DOS 3.3 software and prompts.
- * A black-and-white or color television or monitor (color is preferred; the programs on this diskette are very colorful).

If you have an Apple II, you will need:

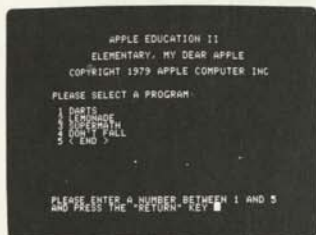
- * An APPLESOFT II BASIC ROM card.

If you have an Apple II Plus, you will need:

- * An INTEGER BASIC ROM card.

If you have installed a Language Card (part of the Apple Language System) in your Apple, you will not need a ROM card, regardless of what kind of Apple you have.

To start using the programs, remove the diskette from its protective paper envelope and insert it in your disk drive (Drive 1 if you have more than one). Now "boot" your system, as described in your DOS manual. (If you have a Language System, see the section "Using BASIC" in your Language System Manual.) The Apple logo will appear momentarily, followed by the Elementary, My Dear Apple main menu:



To use one of the four programs on this diskette, type the number corresponding to that program and press RETURN.

When you finish using a program, it will automatically return you to the main menu. If for any reason a program should stop suddenly and leave you with a blinking cursor and a prompt ("**>**" or "**]**"), press the RESET key in the upper right corner of the Apple keyboard. You will be returned to the main menu. If you ever press RESET accidentally while using the Elementary, My Dear Apple diskette, the program you are using will stop and again you will be returned to the main menu.

When you see the symbol



it means that the next paragraph contains special information that will give you a helping hand.

The symbol



means that the next paragraph describes something you should watch out for: an unusual feature in one of the programs.

CHAPTER 1

THE GAMES

Now that we have the preliminaries out of the way, you're ready to learn how to play the games! There are four of them on the Elementary, My Dear Apple diskette: DARTS, LEMONADE, SUPERMATH, and DON'T FALL. Let's talk about them one at a time.

DARTS

Hurry, hurry, step right up,
Test your skill, and try your luck,
Toss the darts, and then you'll see
If you can win a prize from me!

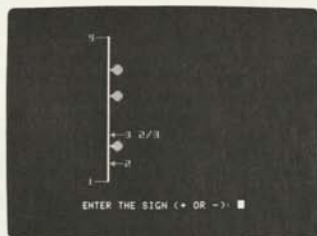
DARTS is a game that tests your knowledge of fractions and number lines. The program will draw a number line on the screen (a different one each time) and place three balloons on it. You try to pop the balloons by throwing darts at them. If you hit all three balloons with four or fewer darts, the program will play a tune for you.

To throw the darts, you must first guess where on the number line each balloon is. The program will give you a hint by showing you the numbers at the top and the bottom of the number line, and one number somewhere on the line. The positions of the balloons are either integers (whole numbers, like 4 or 7), fractions (like $1/2$ or $3/7$), or mixed numbers (like $2\ 1/3$ or $7\ 5/7$). The numbers will all be less than 10, and the fractions will be in halves, thirds, fourths, fifths, sixths, sevenths, eights, or ninths.

In DARTS, you throw a dart by typing the position on the number line where you want the dart to hit. To type a number, you first type the integer (whole number) part. This should be a number between 0 and 9. When you've typed the integer part, press the spacebar. If the number you want to type doesn't have an integer part, just press the spacebar without typing anything.

Now you must enter the fractional part. First, type the numerator (the number on the top of the fraction), and press the spacebar. Then type the denominator (the number on the bottom of the fraction) and press the spacebar again. If the number you want to type doesn't have a fractional part, just press the spacebar instead of typing the numerator.

For example, if you have a number line like this:



and you think the balloon is at $3 \frac{1}{2}$, for DARTS you would type

```
1
3 - (to hit the balloon at 3 1/2)
2
```

(Press six keys: 3 spacebar 1 spacebar 2 spacebar.)

Now watch the screen--the dart will fly across the screen and land on the number line. If the dart hits a balloon, the balloon will pop and the exact position of the balloon will appear on the left side of the line. When the dart lands, the position of the dart will appear on the right side of the number line.

If you hit all three balloons with fewer than four darts, you win! The program will play a little tune for you. Then it will draw a new number line, with three more balloons in different places. If you don't want to throw any more darts, just press the ESC key (on the left edge of the Apple's keyboard).

In SUPERDARTS, the program throws the first dart and you throw a dart by typing the difference between the position of the last dart and the place where you want the next dart to hit. For example, suppose you have a number line like the one in this picture:



You think the balloon is at $2 \frac{2}{3}$, and you threw your last dart at $1 \frac{1}{3}$. For SUPERDARTS you would type

```
1
+1 - (because 1 1/3 plus 1 1/3 is 2 2/3)
3
```

(Press seven keys; + 1 spacebar 1 spacebar 3 spacebar.)

In SUPERDARTS, if you want the next dart to land above the last one, make the number you type positive by typing a plus sign (+) before the integer. If you want to throw the next dart below the last one, make the number negative by typing a minus sign (-) before the integer.

If you press a wrong key (in either DARTS or SUPERDARTS), you can go back and erase the mistake by pressing the forward arrow key. Each time you press this key, you'll back up one step (from denominator to numerator to integer and sign). Once you've erased the mistake, just type the correct number and continue on as if you'd never made the mistake.

TURNING OFF THE MUSIC

This game uses music as much as color, but sometimes an excess of sound (especially if there's more than one person in the room) can be unwelcome. You can change DARTS to make it silent, forgetting about music altogether. It will not even ask any questions about whether you want music or not.

To make this change, choose number 5 (<END>) from the ELEMENTARY, MY DEAR APPLE main menu. You will be faced with a single right-pointing caret (>) and a flashing cursor. Now type

LOAD DARTS

The disk will whir for a moment, then you will be left with the right-pointing caret and a flashing cursor again. Now type

5 MUSIC = NO

If you want DARTS to be silent just this time, type RUN, press the RETURN key, and the program will work normally (except for the lack of sound). Next time you choose DARTS from the main menu, it will be musical again.

If you want DARTS to be quiet every time it's used, then before you RUN it, type
UNLOCK DARTS
SAVE DARTS (remember to wait for the program to be SAVED)
LOCK DARTS

Now every time DARTS is played, it will be silent. This change is permanent, but not irreversible. If, at a later date, you want to restore the music, follow the same steps as above but type
5 MUSIC = YES
instead of the line 5 you used to turn the music off. The game will be melodious again. If you just RUN the program, the change will be temporary; if you SAVE it first, the change will be permanent.



When you make the change to line 5, be sure you type it exactly as it appears here. If you type it wrong, you could damage the program.

LEMONADE

LEMONADE is a game for one or many players. The scene is Lemonsville, California, in the middle of summer. The players choose their corners, mix up a batch of cold, sticky lemonade each morning, and go out to their stands to quench the thirst of passing pedestrians.

You are in charge of one of these lemonade stands. Each morning, you check the weather and decide how much lemonade to make for that day. You post signs around the neighborhood and choose a price for your lemonade. Each evening, you tally up your sales and expenses and find out how much money you've made (or lost). The game will continue until all players agree to quit.

```
HI! WELCOME TO LEMONSVILLE, CALIFORNIA!
IN THIS SMALL TOWN, YOU ARE IN CHARGE OF
RUNNING YOUR OWN LEMONADE STAND. YOU CAN
COMPETE WITH AS MANY OTHER PEOPLE AS YOU
WISH, BUT HOW MUCH PROFIT YOU MAKE IS UP
TO YOU (THE OTHER STANDS' SALES WILL NOT
AFFECT YOUR BUSINESS IN ANY WAY). IF YOU
MAKE THE MOST MONEY, YOU'RE THE WINNER!!
ARE YOU STARTING A NEW GAME? (YES OR NO)
TYPE YOUR ANSWER AND HIT RETURN **)
```

Everybody starts with \$2.00 in cash and some sugar. Lemonade costs 2 cents per glass to make; this cost will rise as the days go by. You can only make as many signs and glasses of lemonade as you've got money to pay for. If you don't sell all the lemonade you make, you lose the money you spent making the glasses you didn't sell.

The amount of lemonade you sell is determined by the price of the lemonade, the weather outside, the number of advertising signs you make, and the number of customers there are. You are not affected at all by the price or sales of the other players' lemonade, nor is today's performance affected by yesterday's sales.

The weather can change from day to day. The longer you play, the more variety you'll see. The warmer the weather is, the more customers you can draw. Sometimes the street department will be working on your street, which keeps your customers away; on cloudy days there is a chance of sudden thunderstorms, which ruin everything for that day.

If you decide to quit, just press ESC (on the left side of the keyboard) after the last player's Lemonsville Daily Financial Report. If you like, you can write down each player's assets and the number of days you have played, and continue the game another time.

SUPERMATH

Now you can practice adding, subtracting, multiplying, and dividing without using flash cards! The program will quiz you on your math skills. It will show you problems and ask you to solve them. SUPERMATH has four parts: addition, subtraction, multiplication, and division. Addition and subtraction can pose problems using numbers up to 999. Multiplication and division use numbers up to twelve.



The more questions you get right, the harder the next question will be. If you want, you can tell the program to quiz you on a certain number, and it'll use that number in every problem. If you can't solve a problem in 3 tries, the program will show you the answer and show you how to solve the problem step by step.

The program lets you solve the addition and subtraction problems column by column, from right to left. Just type the digits of the answer as you figure them out, and press RETURN when you're finished. Many of the harder problems will have you carry or borrow. The biggest number the program will give you is 999, and the result of a subtraction problem is always a positive number.

When you let the program quiz you in multiplication, it will give you problems up to 12 times 12, and ask you to type the answer in all at once, from left to right, and press RETURN.

You can ask the program to let you try division problems, too. You can ask for problems with or without remainders. The biggest number the program will have you divide by is 12, and the program would like you to type the quotient (the answer) in all at once, from left to right, and press RETURN. If you ask for problems with remainders, then the program will ask you what the remainder is after you've typed the quotient.

If you make a mistake while typing the answer to any problem, you can use the left arrow key to back up and retype or erase the wrong numbers. When you've typed the answer correctly, press RETURN. The program will tell you whether your answer is right or wrong. If it's wrong, you'll get another chance to work the problem. If it's right, the program will show you another problem, a little harder than the last one. When you want to stop, just press the ESC key.

DON'T FALL

Here's a game that's like good ol' Hangman, but with a difference. You are walking towards the edge of a cliff. The program will think of a word, and you will try to guess it, letter by letter. Each time you guess a letter that is not in the word, you take another step closer to the cliff. If you don't guess the word in time, you fall!

The program knows more than a hundred words in thirteen different categories. When the program shows you the list of categories it knows, choose one by typing the number next to the name of the category and press RETURN.

The program will then ask you what skill level you want to play at. There are three levels: Easy, Hard, and Expert. The skill level doesn't change how hard the words are, but changes how far away from the cliff you start. For "easy" questions, you start 13 steps away. For "hard" questions, you start 7 steps away, and for "expert" questions you start only 4 steps away from the cliff!

Then the program will ask you whether you want it to play, too. If you let it play, then after you try to guess a word, it will ask you to type any word in the same category as the one you've chosen, and it will try to guess your word. See if you can fool the program and make it fall!

When the program is ready to start, it will show you a picture of a person (that's you) walking towards a cliff, and some dashes, like this:



There's one dash for every letter in the word. Guess a letter that you think is in the word, type it, and press RETURN. If your letter is in the word, the program will take away some of the dashes and put your letter there. If your letter isn't in the word, you take another step closer to the cliff! Too many wrong letters, and you fall! OUCH!

CHAPTER 2

PROGRAMMER'S CORNER: INSIDE DON'T FALL

DON'T FALL is a Shell Game. This means that like the other Shell Games (MR. MULTIPLE, THE MATCH MACHINE, and PROFESSOR TRUE, on the Apple diskette "The Shell Games", Apple part no. A2D0014), the program is actually a "shell" into which you can put your own information quickly and easily with the use of a special editor called the Problem Entry Editor. This editor allows you, in the case of DON'T FALL, to add, delete, or substitute different categories of words simply and painlessly.

DON'T FALL, as well as the other Shell Games, is written in Apple Integer BASIC. Although there is no need for you to have an extensive knowledge of the BASIC programming language, you should at least be comfortably familiar with the editing features available in Apple BASIC. In particular, you should know how to use the left-arrow and right-arrow keys and how to move the cursor around the screen. These features are discussed in the Apple II BASIC Programming Manual. If you are very new to the Apple, you might wish to go over the sections of that manual that cover these topics before proceeding. (Don't let this advice dissuade or frighten you: the Shell Games Problem Entry Editor was designed specifically for new users with about ten hours of programming experience on the Apple.)

If your Apple is equipped with the new Autostart ROM then if your diskette is booted automatically when you turn your Apple on. The Autostart ROM gives you additional ways to move the cursor, as described in either the Applesoft Tutorial or the Autostart ROM Manual.

GETTING INSIDE

To start working on DON'T FALL, first get to the Elementary, My Dear Apple main menu and select option number 5, to END. When you see the Integer BASIC prompt symbol (the right arrowhead, ">") type LOAD DON'T FALL and press RETURN. After the red light on the disk drive turns off, type LIST 9010, 9014

Listing these lines will display a sample of the section of the DON'T FALL program that defines a category. These are the portions of the program with which you will be working when you create lists of your own, so be sure you understand what we are describing.

If you have the original program, here's what you will see:

```
9010 D1$="BIRDS,ROBIN,BLUEBIRD,EAGLE,
CROW,TURKEY,SEAGULL,BLUEJAY,DUCK
,CHICKEN,GOOSE,HAWK,SWAN,ROADRUN
NER,WREN,CARDINAL,SPARROW"
9011 D2$="FLAMINGO,PARROT,CANARY,PENG
UIN,STORK,PARTRIDGE,PEACOCK,OSTR
ICH,PIGEON,DOVE,WOODPECKER,OWL,P
ELICAN,HUMMINGBIRD": RETURN
9012 REM
9013 REM
9014 REM
```

It doesn't matter if you find different lines in your program. Just refer to these lines during the explanation.

Once you understand these five lines and why they look as they do, you will have grasped most of what you need to know to write your own word lists.

ANATOMY OF ALL SHELL GAMES

The general anatomy of the problems is shared by all Shell Games. There are eight principles to that anatomy:

1. The first problem in all Shell Games can always be found beginning on line 9010.
2. All Shell Game word lists occupy exactly five consecutive lines: 9010 to 9014 for the first word list, 9015 to 9019 for the second word list, 9020 to 9024 for the third word list, and so on.

The Shell Games use BASIC strings for holding your word lists. A string, as described in the Integer BASIC Programming Manual, is a series of characters surrounded by quotation marks. For example: NAME\$="FRED", D1\$="FLOWER AND TREES". Strings are simply a convenient way of storing words so you know where to find them.

3. You have five empty strings you can fill for each problem. (You may not need all five.) They are called D1\$, D2\$, D3\$, D4\$, and D5\$.
4. You must use D1\$ for EVERY list. DON'T FALL knows it has finished with a category when it finds a D1\$ with nothing in it.
5. Although you only use as many of the five strings as you need, the strings you use must be in ascending order--that is, D4\$ may not be followed by D3\$. It is not necessary to use each string in order, however: for example, use D1\$, D2\$, and D4\$ for a given problem without using D3\$ at all.

We will talk more about what information you place inside the strings after completing our description of the external structure of the lines. Please keep in mind that we are still discussing things that all the Shell Games have in common.

6. There is always a RETURN statement following the last string used--in this case, a :RETURN on line 9011. But it would be equally effective as
9012 RETURN
or the RETURN could be on line 9013 or 9014--anywhere will do, as long as the RETURN appears after the last string.
7. All line numbers must be used. So--after you have typed your list into the strings, and after you have put in a RETURN statement--if you have any unused lines remaining, enter each of those line numbers followed by REM and a carriage return, just as was done on lines 9012 through 9014. It isn't necessary to type any spaces or other characters after the word "REM".
8. The final feature of all Shell Games is that the last category or "problem" is not really a problem at all. It is a "dummy" RETURN statement consisting of the line number that ordinarily would have been the first line of the next problem, with nothing on it but the word "RETURN". It is called the Data Table Terminator and its line number is always a multiple of 5. Example:
9085 RETURN

This completes our discussion of the external anatomy of all Shell Games. We will now look at the structure of a DON'T FALL word list and then enter a new word list into the program.

THE DON'T FALL WORD LIST

Each word list in the DON'T FALL program consists of the following:

1. The category name as it will appear on the menu.
2. As many words in that category as you can:
 - a. think up, and
 - b. fit in five strings.

The category name and each word in the list is separated from the next word by the special symbol, comma (,).

You should not enter extra spaces. Instead of writing

```
ROBIN, BLUEBIRD, CANARY
you should instead enter
ROBIN,BLUEBIRD,CANARY
```

all scrunched up. If you have a two-word name, like Flying Squirrel, type the space between the two words just as you normally would:
BADGER,FLYING SQUIRREL,FOX

You should think of the five strings, D1\$ through D5\$, as really being one continuous string of characters which happens to be broken into five parts. When DON'T FALL absorbs a word list, it ties the strings together again into one piece, so it really doesn't care how you have divided them. The only rule you must observe, in dividing up your word list among the strings, is that breaks between strings always occur between words, where you would normally be placing a comma. DON'T FALL doesn't care whether you include the comma at the end of the string.

SPECIAL WORD LISTS

There's a special kind of word list that you can use. When you choose to play DON'T FALL with a category that uses one of these special word lists, you will be required to guess all of the words in that category, in order, before DON'T FALL will stop. The Apple doesn't get to play DON'T FALL when you choose a special category, but it will print out a hint for each word you have to guess. To remind you of which categories are special, DON'T FALL will print an asterisk (*) after the names of all special categories. On your DON'T FALL diskette, category 13, SPELLING *, is special.

A special word list is constructed just like a normal word list, except that the name of the category is preceded by a pound sign (#). After each word, you can also add an at-sign (@) followed by the hint you want DON'T FALL to print for that word. Follow the hint with a comma, just as in a normal word list. Notice that the more hints you give, the fewer words you can include.

For example, the DON'T FALL category called SPELLING is constructed like this:

```
9070 D1$="#SPELLING,APPLE@A GREEN FRU
IT,SALAMANDER@A SLIMY LIZARD"
9071 D2$="ICE CREAM@A COLD DESSERT,CH
AIR@YOU SIT ON IT,LETTERS@POSTIME
N CARRY THESE,DOG@MAN'S BEST FRI
END"
9072 RETURN
9073 REM
9074 REM
```

The pound sign in front of the title SPELLING tells DON'T FALL that this is a special word list. A word is separated from its hint by an at-sign; each word-hint pair is separated from the next by a comma. When DON'T FALL asks you to guess a word from a special category, after it asks you to PLEASE GUESS A LETTER, instead of saying AND PRESS THE 'RETURN' KEY it will print the hint that's associated with the word. You have to guess all of the words in the category before you can stop playing.



Teachers might want to use these special categories to store weekly spelling lists, with which students could quiz themselves while playing DON'T FALL.

THE PROBLEM ENTRY EDITOR

Now you are going to learn how to enter a new category. You are going to create a category named Rivers.

Inside DON'T FALL is a hidden program called the Problem Entry Editor, designed to make problem entry as simple as possible. If you haven't LOADED the DON'T FALL program, do so now, as described above in the section "GETTING INSIDE". To get to the Problem Entry Editor, type RUN 2

ENTERING A NEW CATEGORY

You are now in the Problem Entry Editor. Select

1) ENTER A NEW PROBLEM

from the menu. The first thing that occurs is that DON'T FALL leafs through all the first lines of the existing categories, looking for the Data Table Terminator line: the first line of a problem (an even multiple of five) containing only a RETURN statement. The program knows it has found the Data Table Terminator because D1\$, which was emptied just before jumping to the problem, is still empty on return. (This is why it is always necessary to use D1\$ in every word list.)

The second thing that usually happens is that the Apple will ask you to copy over a couple of lines, pressing the RETURN key at the end of each, as seen in this photograph:



The cursor will be positioned at the beginning of the first line. Just use the forward arrow key to move across each character, until you reach the space just before the black-on-white arrow. Then press the

RETURN key. The cursor will disappear for a moment and then appear at the beginning of the next line. Step through the second line using the forward arrow key and again press the RETURN key just before the black-on-white arrow.

As the new category will be entered at the end of the data table, beginning on the line now occupied by the Data Table Terminator, you need to write a new Data Table Terminator five lines further ahead. That is what you just did by copying over those lines with the forward arrow key.

The reason the Problem Entry Editor doesn't always stop and ask you to copy lines is that there may already be a RETURN statement five lines ahead. Data Table Terminators tend to get left lying around rather casually. The program doesn't mind--as long as there's one, there may be a hundred.

After copying over the lines, you will be presented with a choice at the bottom of the screen:
<ESC> FOR EDIT, <RETURN> TO MENU

If you reach this point accidentally, you can press the RETURN key to return to the Problem Entry Editor menu. Since you have come this far intentionally, press the ESC key. Here's what you'll see:



(If the line numbers do not correspond, you simply have a version of the program with more or fewer categories already entered. The specific line numbers you see are unimportant.)

The Problem Entry Editor has laid out the five strings in their maximum size and placed the cursor at the beginning of the first line. You're given a scant four lines for each string, at the end of which is the close-quote, followed by a white rectangle. The second quotation mark is there to indicate the absolute maximum length the string may be; it is rare that your word list will fit exactly. You may enter less than the maximum length by simply closing the string: type a quotation mark, then press the RETURN key. In no case may you go beyond the quotation mark. If you do, you will receive a
*** TOO LONG ERR
message and have to re-enter the line.

If you use all five strings, then the last string must be followed by the RETURN statement; this is why it is shorter than the other four.

Keep in mind that displaying these maximum string sizes and the RETURN statement on the last line is just a way of reminding you of the limitations and requirements. In actual operation you will usually need fewer than the maximum number of strings and less than the maximum string length, and you may put the RETURN statement anywhere as long as it follows the last string you actually use.

You may now proceed to enter the new category. The title to appear on the menu is

RIVERS
and you will want the following list of rivers: Nile, Mississippi, Amazon, Yangtze, Thames, Volga, Danube, Po, Ganges, Mekong, Colorado, Columbia, Congo, Zambesi, Orinoco, Irrawaddy, Limpopo.

To enter the list, first press the forward arrow key repeatedly to move the cursor just beyond the first quotation mark on the first line. Then start typing in the list, starting with the title, Rivers.

For example, you might enter:
9075 D1\$="RIVERS,NILE,MISSISSIPPI,AMA
ZON,YANGTZE,THAMES,VOLGA,DANUBE"

Since you didn't fill the entire allotted space in the string, be sure to include the closing quotation mark after the last character and press the RETURN key. If the cursor isn't at the beginning of the next numbered line, press the backward arrow key repeatedly. This will move the cursor down one row at a time until it is on top of the 9 in 9076. Remember this technique: whenever the cursor is at the left edge of the screen and you wish to move it down the screen a line at a time, use the backward arrow key.

Now that you have positioned the cursor on the line you want it on, space it forward with the forward arrow key until it is just past the opening quotation mark on the line. Then continue typing:

9076 D2\$="PO,GANGES,MEKONG,COLORADO,C
OLUMBIA,CONGO,ZAMBESI"
9077 D3\$="ORINOCO,IRRAWADDY,LUMPOPO"

(Notice that the last river name is misspelled. Don't worry; type it this way and you'll find out later how to correct it.)

9078 RETURN
9079 REM

You could have crowded almost the entire category into the first string, but you may wish to come back at a later time and lengthen the title, add new river names, or make other changes that require increasing the amount of material. This will be far easier to do if you have left some room.

Normally, the next thing you would do would be follow the screen's advice to RUN 2, but don't do so now; we haven't covered any of the other features of the Problem Entry Editor yet.

Check over the lines you have entered for accuracy and then type RUN to RUN the program and see your work revealed. If there are any problems, you may wish to refrain from correcting them until you have read the next section on editing.

As you can see, the Problem Entry Editor makes new problem entry easier by finding the proper location for the next problem, doing some of the typing for you, and providing end-of-string and don't-forget-the-RETURN-statement reminders. But it takes time for the Problem Entry Editor to find the location of the next new problem. If you are entering an entirely new set of categories all at once, you could delete the entire data table by typing

```
DEL 9010,9999
```

and you may find it a lot faster not to use the Problem Entry Editor. If you are doing all new categories, you know on which line the next category will begin, and it can often be faster to enter it "by hand", perhaps using the AUTO command to supply you with line numbers. (See the Apple II BASIC Programming Manual). Try it both ways and see which you prefer.

EDITING AN EXISTING CATEGORY

If you have been following along on the computer, you will now have a river named the Lumpopo. Unfortunately, no one else in the world has ever heard of such a river. The proper spelling, as Rudyard Kipling would quickly tell you, is Limpopo. The Problem Entry Editor makes it easy to fix this mistake. Type

```
RUN 2
```

to enter the Editor. Select

```
2) EDIT CURRENT PROBLEM
```

from the menu and press the RETURN key.

The category you entered earlier is displayed and you are given the option of entering edit mode or returning to the menu. Since you wish to edit, press the ESC key, as indicated at the bottom of the screen. The screen now appears like this:



The lines have been moved over so that there are no extra spaces on the left, relieving you of having to leapfrog over them. The cursor has been placed at the beginning of the first string. You may enter as much text as will fit between the beginning of each string and the small white rectangle--in fact, the cursor can appear on top of the first block of the small rectangle when you press RETURN. The word list you are entering falls safely short of these maximums.

Since you wish to edit the third line, use the backward arrow key, as described above in the section "ENTERING A NEW CATEGORY", to move the cursor down one row at a time until it is on top of the 9 in 9077. Then press the forward arrow repeatedly until the cursor is resting over the U in LUMPOPO. Type the letter I in place of the U and then press the forward arrow repeatedly until it is on top of the first space following the quotation mark. Then press the RETURN key. The new line has now replaced the original.

You may now either type

```
RUN 2
```

or backspace to the bottom line and copy over

```
RUN 2
```

with the forward arrow. Either way, press the RETURN key and you will be returned to the Problem Entry Editor menu.

During the time that you are editing a problem or entering a new one, you are not being controlled by the Problem Entry Editor. The Editor has set up the screen and printed reminders to guide you, but you are essentially talking directly to the Apple. At this point, it is important to know how to use the Apple editing features: the forward and backward arrow keys and the various ESC sequences--ESC-A, ESC-B, and so forth. With them, you can do extensive editing of problems with speed and ease. Take the time to practice until you become proficient.

If, while entering or editing data, you find the screen becoming cluttered, or you get an error message that is covering up part of what you are doing, type
RUN 2
again and select
2) EDIT CURRENT PROBLEM
from the menu. This will clear the screen and redisplay the lines you have already entered.

BACK TO THE MAIN PROGRAM

Now you can check that the word list is working properly. To do so, select

3) GO TO GAME
from the Problem Entry Editor menu. This selection will cause the program to run in the normal way, beginning at the category selection, except that you have the option of instantly returning to the Problem Entry Editor at any time simply by pressing the ESC key. You may now play the game to confirm that all is well, or you may just select a category, skill level, and number of players, and then press the ESC key to return to the Problem Entry Editor.

We will not be referring to this newly entered word list again. If you are tired of reading and will want to come back later, you need not keep the program in the computer any longer. You can LOAD the original version again later. If you wish to keep this version of the program containing the new category, you may now SAVE it the same way you SAVE any other program you have written.

SEARCHING FOR A CATEGORY

One of the prime reasons for the creation of the Problem Entry Editor was the difficulty of trying to find a particular word list somewhere in the middle of the data table. Listing forty or fifty problems in order to find the one with a typographical error in one word can be most tedious. There are two features of the Problem Entry Editor that were designed to alleviate this problem.

First, in all the Shell Games, the Problem Entry Editor is always aware of where you are in the program. Exit the Problem Entry Editor by selecting

7) END
from the menu and type
RUN

Select category 10 from the menu and begin guessing the word. You might be playing this game and suddenly discover that BAGPIPE is spelled BIGPIPE. Stop the program by typing a CTRL-C, then type
RUN 2

The Problem Entry Editor indeed knows you are working on category 10 and is prepared to set up the program lines in that category for editing, as you can see by the second selection on the Problem Entry Editor menu.

Of course, there will be occasions when you do not wish to edit the particular problem you have been displaying. So the Problem Entry Editor has a search feature to allow you to find the word list you do wish to work on.

Select

4) SEARCH FOR PROBLEM TO BE EDITED

You may begin your search from one of three places: the beginning, the current problem, or the end of the table. When looking for a particular word list, make your best guess as to which of these three locations is closest to the word list you're looking for, and select that option. Beginning your search from

3) THE END OF THE DATA TABLE

is the most fun, so select that now.

The Problem Entry Editor just went forward from the previous current problem to the the Data Table Terminator. The line you are looking at is the Terminator: a line number that is an even multiple of five with only a RETURN statement after it. Pressing the backward arrow key will move you back to the last word list.

By using the backward and forward arrow keys to move backward and forward through the categories, coupled with the REPT key for speed, you can quickly find any word list. Notice, however, that for speed and clarity only the first line of each list is displayed. Once you have found the word list you wish to edit or display, press the RETURN key. This will take you back to the main Problem Entry Editor menu. The Problem Entry Editor will now consider the word list you were viewing at the time you pressed RETURN to be the new current problem and you can view or edit it.

VERIFYING THE DATA TABLE

As simple as the structure of the word lists may now seem, it is quite easy when entering a lot of categories to make mistakes, such as not terminating each list with a RETURN statement or forgetting the Data Table Terminator. Some of the errors created by these minor slips can be very difficult to understand. Not having a RETURN statement at the end of a problem, for example, will cause the next problem to occur twice! Since it is the computer's job to be concerned with such matters, the task of checking for errors in the form of the data table has been automated. Your Apple will do it for you if you select
5) VERIFY DATA TABLE FORMAT

If there are any problems, the Problem Entry Editor will announce exactly what they are and help you correct them, usually printing out the corrected lines and asking you to copy them with the forward arrow key, as you did earlier in entering a new word list. After you have made the correction, the Problem Entry Editor will verify that you have done it properly, asking you to do it again if it is wrong.

The only type of error to which the computer cannot dictate a correction is a missing RETURN statement. If you have forgotten the RETURN statement at the end of a word list, the Problem Entry Editor will tell you so and display the list that lacks the RETURN statement, leaving you in edit mode. You may enter a RETURN statement following the last string used and thus resolve the problem. Afterwards, return to the Problem Entry Editor and verify that the data table is now correct.

When all the word lists are fixed up, the Editor will give your data table a clean bill of health and announce that you should press the RETURN key to return to the menu.

REPLACING OLD CATEGORIES

To replace one list with another, search for the list you wish to replace, select

2) EDIT CURRENT PROBLEM

and delete the lines of the old word list. Then re-enter the Problem Entry Editor and select

1) ENTER A NEW PROBLEM

The empty strings will be displayed and you may proceed to enter your new list. If this is your first time through the Problem Entry Editor procedure, remember that you must actually enter a new list—you may not leave a gap in the data table! If you do, you are breaking two of the cardinal rules of the Shell Games: all Shell Game problems occupy exactly five lines, and all line numbers must be used.

INSERTING AND DELETING LISTS

Although it is possible to insert and delete problems from Shell Game data tables, these are not tasks for a brand-new programmer. Unless you are fairly experienced, it is suggested that you skip this section. You can make full use of the Shell Games without doing any insertions and deletions.

Both insertion and deletion depend on the Integer BASIC RENUMBER program. This program resides in the Programmer's Aid #1 ROM card and can also be found on an Apple Software Bank diskette, Contributed Programs Volume 5, both available from your Apple dealer. Instructions for using the RENUMBER program are provided with these products.

For each new word list you wish to insert, renumber the data table starting from the point where the new category will begin and ending at line number 9999 with an increment of one, freeing five lines for your insertion. To delete a word list, delete its lines and then renumber starting from the beginning of the next list and ending at

line number 9999 with an increment of one, using up the five lines that are left free. For example, to insert a list before category 5, which currently resides from line 9030 to 9034, you would give these commands to RENUMBER:

```
CLR
START = 9035
STEP = 1
FROM = 9030
TO = 9999
CALL <central memory address for renumbering a program>
```

Conversely, to delete category 5, enter:

```
DEL 9030, 9034
CLR
START = 9030
STEP = 1
FROM = 9035
TO = 9999
CALL <central memory address for renumbering a program>
```

The specific CALL address you will use depends on which version of RENUMBER (RAM or ROM) you are using and, in the case of the RAM version, where you have located the program. Consult the instructions supplied with the RENUMBER program for the address of the CALL.

This insertion and deletion scheme depends upon each and every line in the data table being used, so compression won't occur. If only four lines were used for a category, the next category's first line would suddenly become the last line of the previous list, creating chaos and confusion. This is the reason for requiring each line number to be used and for the menu selection

6) VERIFY TABLE FORMAT TO ALLOW RENUMBER

which, in addition to checking everything covered by selection 5, also confirms that all line numbers are used. If any unused lines are discovered, the program will print the lines that need to be corrected and ask you to copy them.



WARNING: It is vital that you do this verification before attempting to insert or delete, as "decompression" is frustrating and time-consuming.

LEAVING THE EDITOR

Selecting 7) END is the preferred way to exit from the Problem Entry Editor, as it will remind you to save the program if you have made any changes.



WARNING: Not saving your program will result in the loss of all the changes and additional word lists you have entered.

AN IMPORTANT FRINGE BENEFIT

When a Shell Game is ended, it is deleted and the INDEX program is RUN. This is most distressing if you have changed one or more problems but have not yet saved the new version, as it will suddenly cease to exist. When you type

RUN 2

the first thing that occurs is that this return-to-INDEX routine is disabled so you can save at your leisure. When you are completely through with the program, you may type

RUN INDEX

to reconnect the INDEX program and return to normal operation.



There are also other more subtle protective measures taken to protect your work from harm, so it is a good idea always to enter the Problem Entry Editor at least momentarily before working on the program, even if you will be entering problems "by hand".

INDEX

A

adding 8
Address 22
Apple Integer BASIC 10
asterisk 13
at-sign 13
AUTO 17
Autostart ROM 10

B

BASIC 10
Renummer 21
Strings 11

C

CALL address 22
category 10

D

DARTS 1, 3
Data Table
End 20
Format 20
Data Table Terminator 12, 14, 15, 20
DEL 17
DELETE 17
Deleting lists 21
denominator 3
dividing 8, 9
Dn\$ 11, 12
DON'T FALL 1, 9, 10, 11
word list 12
DOS 2

E

EDIT CURRENT PROBLEM 17
END 22
END OF THE DATA TABLE 20
ENTERING A NEW CATEGORY 18
Error 15, 21
ESC 4, 8, 9, 15, 18, 19
EXIT see Leaving

F

Format 20
Fractions 3

G

GO TO GAME 19

H

Hangman 9
hint 13

I

INDEX 23
Inserting lists 21
Integer BASIC 10
Integers 3

J

K

L

LEAVING the Editor 22
LEMONADE 7
line numbers 15
LOAD 14, 19
LOCK 6

M

main menu 5, 10
mixed numbers 3
multiplication 8, 9
Music 5

N

number lines 3, 4
numerator 3

O

P

pound sign 13
Problem Entry Editor 10, 14, 15,
17, 20

Q

QUIT see Leaving
quotient 9

R

Remainders 9
Remember 21,22
REPLACING OLD CATEGORIES 21
RESET 2
RETURN statement 12, 14, 15, 16,
17, 20, 21
RUN 17, 19, 23
RUN 2 14, 17, 18, 19, 23

S

SAVE 6, 19
Search 20
SEARCH FOR PROBLEM TO BE EDITED 20
Shell Games 10, 11, 12, 21
Spelling 13, 14
Strings 15
Subtracting 8
SUPERDARTS 4
SUPERMATH 8

T

Table Format verify 22
TOO LONG ERROR 15

U

UNLOCK 6

V

Verify 20, 22
Table Format 22
Data Table 20

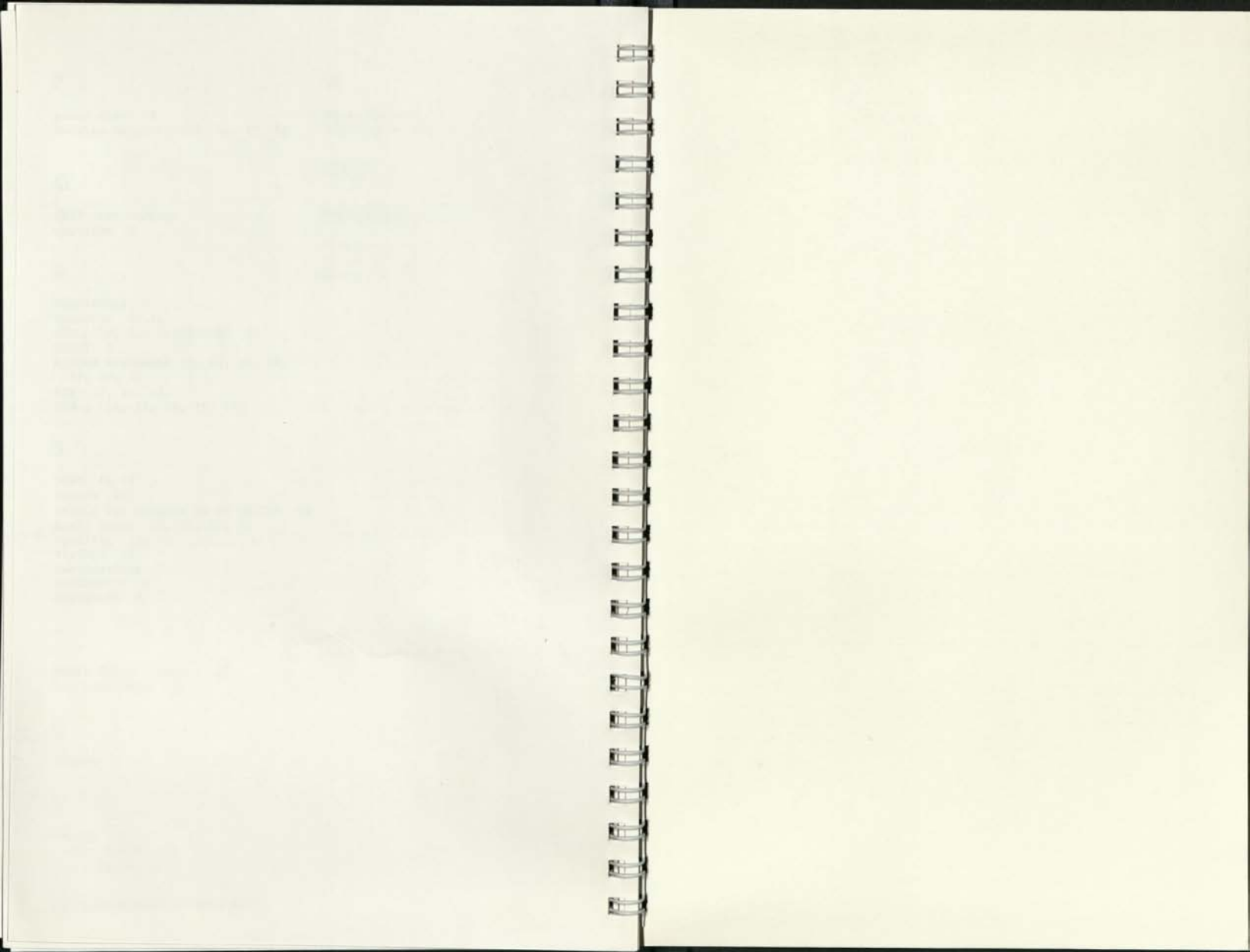
W

Whole numbers 3
Word Lists 12, 13, 16

X,Y,Z

SYMBOLS

* 13
13
@ 13





10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

APPLE USER INPUT REPORT

for the Apple II and the Apple II Plus



Apple Software Support
 Apple Computer Inc.
 10260 Bandy Drive
 Cupertino, CA 95014

DATE _____

CONTROL # **654988**

INTERNAL
 USE ONLY

Apple Owner

Apple Dealer

Name: _____

Person to Contact: _____

Company: _____

Store Name: _____

Address: _____

Address: _____

City/State/Zip: _____

City/State/Zip: _____

Home Phone: () _____

Phone: () _____

Work Phone: () _____

Service Center # _____

Received

Response
 Sent

This is a report of a problem with an Apple product.
 suggestion for changing or improving an Apple product.

Describe your Apple hardware set-up:

Memory Size

Apple II Serial No. _____
 (appears on bottom side of Apple)

16K 48K

Apple II Plus 32K Other _____

List any Cards plugged into the slots inside the Apple (e.g., Disk Controller Cards, Language Card) and the accessories (e.g., Disk II Drives) connected to them:

Slot 0* _____ Slot 4 _____

Slot 1 _____ Slot 5 _____

Slot 2 _____ Slot 6 _____

Slot 3 _____ Slot 7 _____

*Note: These are Slot Numbers, not Disk Drive numbers or Pascal Volume Numbers.

Type of Video Display used: _____

Any special P/ROMs: _____

Game I/O Devices (e.g., game paddles, light pen, joy stick): _____

Other: _____

Describe any modifications to the Apple hardware: _____

Disposition

(over)

List software products (the program, the operating system and language) in use at the time of problem, with their version numbers, if applicable:

<u>Exact Name of Software Product</u>	<u>Version No.</u>	<u>User Registration No. (If Applicable)</u>

If the problem relates to documentation, enter the appropriate code(s) and page number(s) in the columns below.

- | <u>CODE</u> | <u>CODE</u> |
|---|--------------------------------------|
| D1 typographical error, word omission, etc. | D5 information is incorrect |
| D2 format is confusing | D6 inconsistency with another manual |
| D3 information is presented unclearly | D7 inconsistency with software |
| D4 information is missing or insufficient | D8 inconsistency with hardware |

Be sure to include the name of the manual and its number (030-____ - __), which is found inside the front cover, in your explanation.

- Describe the problem/suggestion in the space below (additional sheets may be attached).
- Be specific about the conditions of the event (e.g., which keys were pressed prior to problem).
- Attach program listings and/or sample runs if applicable and possible. If your data diskette demonstrates the event, include a copy of it.
- Describe any remedies you may have found for the problem.

Code	Page	Description of Hardware/Software/Documentation Problem or Suggestion

