people's computers
PCC
Box E
Menlo Park, CA 94025

Second class postage
paid at Menlo Park,
California 94025 and
additional entry points

93065JOHNSB03 2227

BRYON JOHNSON
356 LAGUNA TERR
SIMI VALLEY, CA 93065

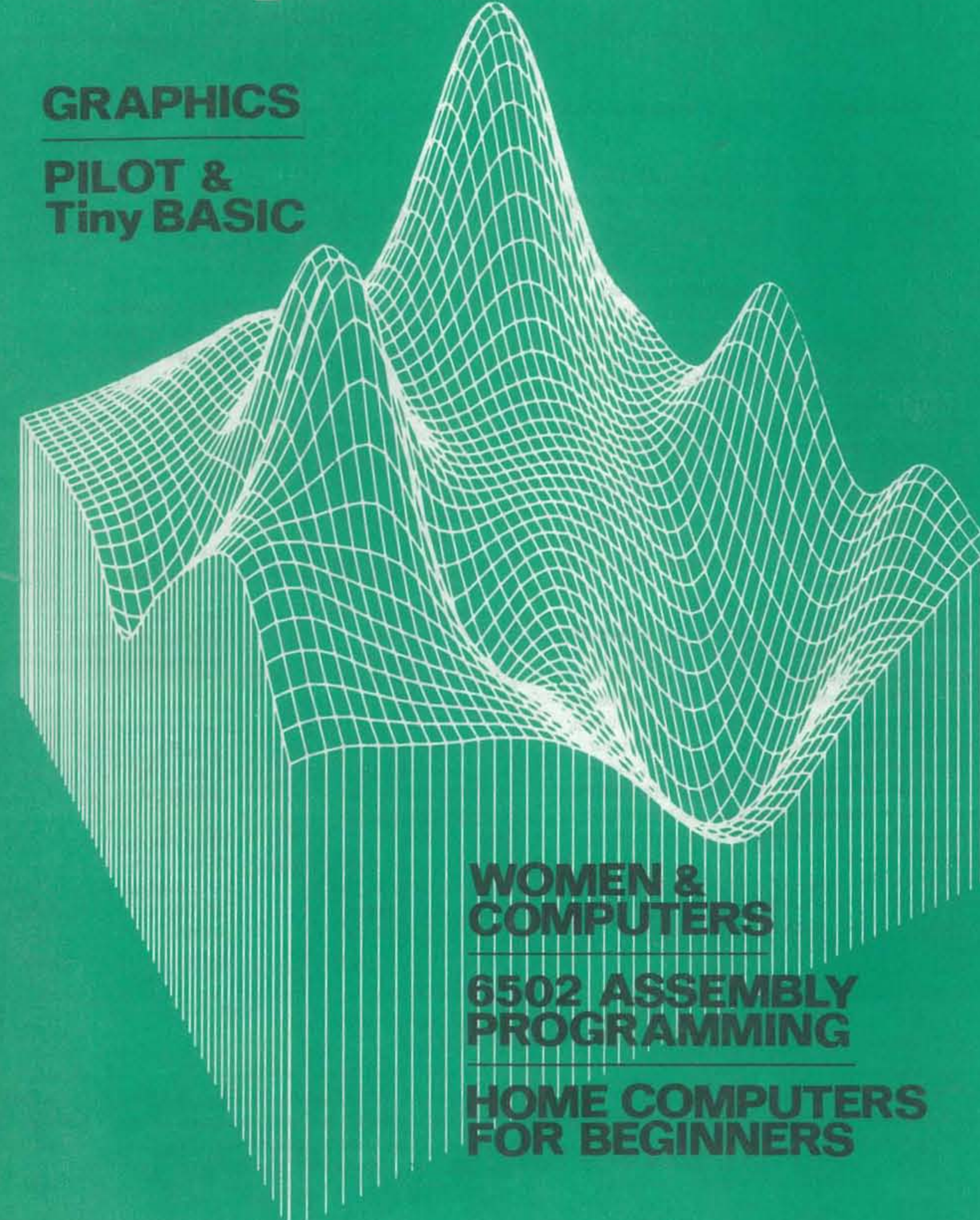
people's computers

$1.50

VOL. 5 NO 6 MAY-JUNE 1977

GRAPHICS

PILOT & Tiny BASIC



WOMEN & COMPUTERS

6502 ASSEMBLY PROGRAMMING

HOME COMPUTERS FOR BEGINNERS

SUBMITTING ITEMS FOR PUBLICATION

LABEL everything please, your name, address and the *date*; tapes should also include the program name, language and system.

TYPE text if at all possible, double-spaced, on 8½ x 11 inch white paper.

DRAWINGS should be as clear and neat as possible in black ink on white paper.

LISTINGS are hard to reproduce clearly, so please note:

- Use a new ribbon on plain white paper when making a listing; we prefer roll paper or fan-fold paper.
- Send copies of one or more RUNS of your program, to verify that it runs and to provide a sense of how things work -- and to motivate more of us to read the code. RUNS should illustrate the main purpose and operation of your program as clearly as possible. Bells, whistles and special features should just be described in the documentation unless they're particularly relevant.
- Paper tapes of both the program and runs can provide us with a way to make our own listing if we need to. Then, if you give us permission, we can let CCC (Community Computer Center) sell your program cheaply via paper tape, to further the spread of inexpensive software. Finally, if we are so lucky as to have access to a system on which your program runs, we can try it out ourselves.
- Make sure your code is well documented — use a separate sheet of paper. Refer to portions of code by line number or label or address please, not by page number. When writing documentation, keep in mind that readers will include beginners and people who may be relatively inexperienced with the language you're using. Helpful documentation/annotation can make your code useful to more people. Documentation should discuss just which cases are covered and which aren't.
- If you send us a program to publish, we reserve the right to annotate it (don't worry, we won't publish it if we don't like it).
- Last but not least, please try to limit the width of your listings: 50-60 characters is ideal. Narrow widths mean less reduction, better readability, and better use of space.

SUBSCRIPTIONS

U.S. Subscriptions
- ☐ $8/yr. (6 issues)
- ☐ $15/2 yrs. (12 issues)
- ☐ Retaining subscription @ $25 ($17 tax deductible)
- ☐ Sustaining subscription @ $100+ ($92+ tax deductible)

Foreign Surface Mail
- ☐ add $4/yr. for Canada
- ☐ add $5/yr. elsewhere

Foreign AIRMAIL
- ☐ add $8/yr. for Canada
- ☐ add $11/yr. for Europe
- ☐ add $14/yr. elsewhere

Back issues, $1 each; indicate Volume and Issue number, how many copies of each. An order card is at the center of the magazine.

Vol. 1, Nos. 1, 2, 3
Vol. 2, No. 5
Vol. 3, Nos. 1, 4
Vol. 4, Nos. 3, 4, 5, 6
Vol. 5, Nos. 1, 2, 3, 4, 5

Foreign Distributors of *People's Computers*

Vincent Coen
LP Enterprises
313 Kingston Road
Ilford, IG1 1PJ
Essex, UK

Comicro AG
Badenstrasse 281
CH-8003 Zurich
SWITZERLAND

Pan Atlantic Computer Sys.
Frankfurter Strasse 78
D61 Darmstadt,
WEST GERMANY

Home Computer Club
1070-57 Yamaguchi
Tokorozawa, Saitama, JAPAN

Kougakusha Publ. Co., Ltd
Haneda Biru 602, 5-1
2-Chome, Yoyogi
Shibuya-Ku, Tokyo 151
JAPAN

Computer Age Company, Ltd
Kasumigaseki Building
3-2-5 Kasumigaseki
Chiyoda-Ku, Tokyo 100
JAPAN

People's Computers is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025. People's Computer Company is a tax-exempt, independent, non-profit corporation, and donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points.

people's computers

VOL 5 NO 6
MAY-JUNE 1977

STAFF

EDITOR
Phyllis M. Cole
PRODUCTION MANAGER
Meredith Ittner
ARTISTS
Ann Miya
Judy Wasserman
TYPISTS
Maria Kent
Marilyn Newton
Renny Wiggins
CIRCULATION
Beverly Robinson
DRAGON EMERITUS
Bob Albrecht

RETAINING SUBSCRIBERS

George Bowie III
The Computer Corner, Harriet Shair
Daniel Dick
Bill Godbout Electronics
Mark S. Elgin
Dexter Fletcher
John B. Fried
Dick Heiser
Brian W. John
Lt. Stan Jurgielwicz
John R. Lees, Jr.
John C. Lilly
James Muysenberg
Frank Otsuka
Bernice Pantell
Larry Press
John Rible
Joe Weisbecker

SUSTAINING SUBSCRIBERS

Algorithmics Inc., Bruce Cichowlas
Don L. Behm
Paul, Lori and Tom Calhoun

As ever, thanks to the many many folk who supported our effort in putting this issue together.

CONTENTS

SPECIAL FEATURES

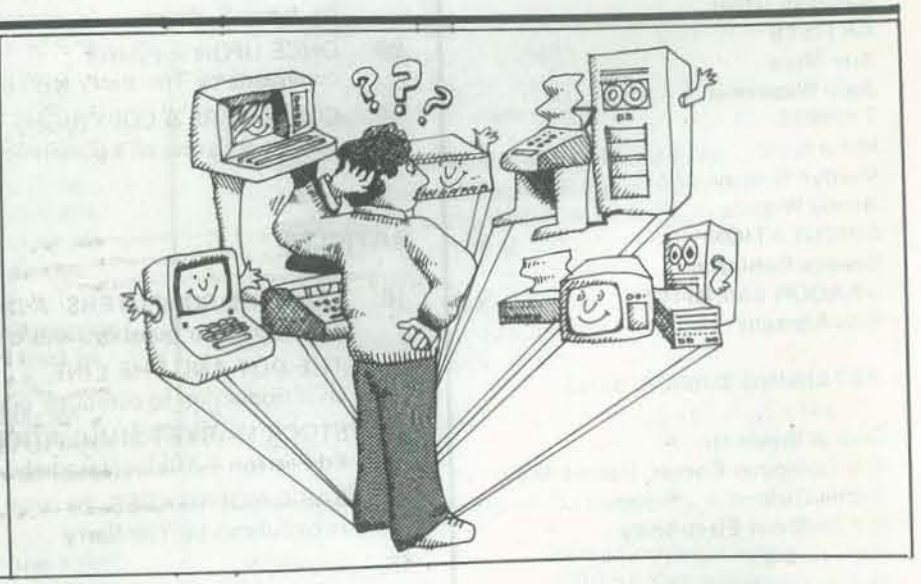
ARTICLES

REGULAR STUFF

Home Computing:
AN INTRODUCTION FOR NOVICES

BY JAMES S. WHITE

At last! An introduction to computers that assumes no prior technical knowledge! The article is preprinted with permission from the Proceedings of the First West Coast Computer Faire, where it appears under the title, 'An Introduction to Computing to Allow You to Appear Intelligent at the Faire.' Jim White has worked with a variety of newcomers to microcomputers. He is the author of Your Home Computer: An Introduction to Personal Computing, available for $6.00 through the People's Computer Company bookstore. The first two illustrations accompanying this article appear in his book.



INTRODUCTION

Welcome to home computing! The time has arrived for a computer in the home of almost everyone who wants one.

During the past few years, expensive, complex computers in industry and commerce have helped you by cutting costs and raising the quality of many products and services you use. Now, computers have been designed for home use and their prices have been drastically cut. Personal computers are available today at prices similar to other major household purchases.

Thousands of Americans are enjoying their own computers now. You can select a computer for your home, a computer that will be easy for you to use, even if you have no technical experience. This paper describes how you can benefit from your own computer.

Learning about computers can be a little confusing because many different kinds of things are sometimes called computers. A computer can not be consistently identified by its size, appearance, construction or use. None of these features are characteristics of computers in general. Computers are electronic devices built according to certain technical rules, which we won't discuss here.

However, all types of computers do have some common characteristics that make them different and much more powerful and helpful than many other things we use. We will discuss some common characteristics of today's popular types of home computers, characteristics that are important to computer users and that make computers especially helpful to us.

COMPUTERS ARE TOOLS

We use tools to extend our capabilities and to multiply our natural powers. We use tools to help us do things we can't alone, or so we can do jobs easier, quicker, or better than we can alone. Computers help us in all these ways.

MENTAL TOOLS

Computers help our minds. Rather than helping our muscles, as most tools do, computers multiply our intellectual power and capabilities. Computers are extensions of our minds. Computers are tools we can use for greatly increased mental achievement and enjoyment. Because our mental activities are much less limited in potential benefits than our physical activities, the helping potential of computers is much greater than the potential of other tools.

PROGRAMMABLE TOOLS

Computers follow people's instructions and do nothing unless instructed to do so. Computers are useful as extensions of our minds primarily because they faithfully do just as we instruct. Your computer will follow your instructions. Your computer's ability to follow a complex set of instructions and thus do just what you specify gives it power far beyond any other kind of tool. We don't know the limits of the ultimate potential of computers because we don't know the limits of the ultimate capacity of men and women.

If you examine a computer in detail, you see that it recognizes only a few instructions that can do only certain limited things. These instructions allow a computer to, for example:

1. Perform mathematical calculations
2. Store a large amount of information, and select a particular piece of information to meet a specific need
3. Evaluate a vast number of alternatives to determine the best solution to a question, even a question unrelated to mathematics.

However, the instructions a computer recognizes are so powerful and so varied that a person can combine them to instruct his computer to do almost anything as an end result.

A set of instructions you give to your computer is a *program*. When you instruct your computer, you are *programming* it.

A computer's program can be changed easily and quickly. A complete program can be changed in as little as a fraction of a second. However, the work or play of composing a program the first time usually takes a person several hours or longer.

CONTROL TOOLS

Computers can help us by controlling other tools and machines. A computer can cause a much larger tool to do something in a certain way and at certain times, as programmed. This capability extends the power of a computer far beyond the things it can do directly.

Our minds are control devices which control our muscles. We thereby control the most powerful of machines, the generation and use of power, and many natural events.

So, although computers function intellectually, their direct benefits extend far beyond the intellectual. By helping us to "work smarter," computers can be a way to physical accomplishment that is better than just working harder.

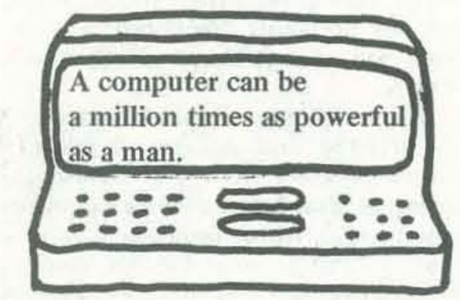
MACHINE TOOL

A computer is a machine. A machine is generally a tool that can operate by itself, doing a desired job, without always needing the personal control or constant attention of a person. A computer can follow its assigned program with little or no need for the further assistance of a person.

A machine used in a home is generally called an appliance. A computer can be a home appliance, a machine that can continue working to help us while we do things that are more useful or enjoyable.

VERSATILE TOOLS

Computers can be extremely vertatile. One computer can do, and be, many different things. A computer's versatility results from two other characteristics. The first is a computer's previously discussed programmability, the fact that you may instruct your computer to do whatever you want.



The second reason a computer is versatile is that it is a system. A system consists of various parts, each doing its own job. All of a system's parts operate together, supporting each other as do the various organs in our body.

A computer can be composed of a few or many different types of parts; each can have widely varying characteristics. The parts you choose to assemble your computer are those parts that together will give the resultant system the capabilities to best do what you want it to do, at a price you can afford. You can change these parts as you want your computer's capabilities to change.

The wide range of possible computer characteristics also results from the fact that a computer, as a control device, is often combined with other devices. A large variety of devices may be combined so the resulting system can have a wide variety of characteristics. A computer generally remains the key part of the resulting combination, so all these types of systems are often called computers.

Computers are so versatile that a computer can do, or control, any rational or mechanical process, activity, or happening. What a computer can do is essentially limited only by the skill and imagination of the person(s) who build and program the computer.

A particular computer must be equipped with the power and accessories appropriate for the job it will do. Power usually isn't a serious constraint; a computer that is half as powerful may just take twice as long to do a given job. Even computer accessories are usually versatile, and many can be used for a wide variety of different kinds of jobs.

PRECISE TOOLS

Computers can be perfect. A properly programmed and operated computer makes so few errors in end results that a computer mistake may never occur during years of doing a particular job. Computers do occasionally make mistakes, typically one in many millions of operations. However, computers can, and many do, check their own work so that a mistake is not given to a human user.

Most "computer mistakes" occur because computers do exactly as they are instructed. Many people find it difficult to give correct instructions to handle complex problems. The computer that follows incorrect instructions will produce an incorrect result, and is often incorrectly blamed.

FAST TOOLS

Computers are extremely fast. A typical home computer can do 500,000 of its basic operations per second. Home computers often move information as fast as physically possible—at the speed of light.

This great speed also gives computers great power. Because computers are so fast, they are very useful even for those jobs that they can't do efficiently. Computers just go ahead in their roundabout way, doing many, many operations to provide a useful result. A typical computer that must do a million operations to complete a particular job may be done in two seconds, which is often sooner than any alternative.

How powerful are computers? A computer can be a million times as powerful as a man. That's almost beyond comprehension, so let's consider an example.

The great pyramid of Egypt is generally accepted as the world's most costly single structure, in terms of man years. According to the most commonly accepted estimates, 100,000 men worked 20 years. In total, two million man years was required to do this job.

Building pyramids is out of style today. Our world is oriented towards numbers, such as dollars. So let's think about two million man years of working with numbers or calculations.

The first common home computers were delivered about two years ago. One of these computers, working for these two years, could have done calculations that would take one million men working with pencil and paper 24 hours per day, 365 days per year, for 2 years. You can buy such a computer, or one much faster, in your local computer store.

PERSONAL TOOLS

Despite their great power, computers can be personal tools. You can select your own computer to help you, and to expand your own capabilities. More than 10,000 Americans have their own computers; more than 100,000 Americans are using computers as personal tools, to make their work easier and their play more fun.

We should consider a couple of philosophical points here. Despite its great power, a computer can no more do anything by itself than can any other machine. Computers are constructed, and instructed, by man, as tools. A computer is and does only what it is made to be and do by man.

Further, as we plan to use computers, we should consider their strong and weak points, just as we do for other tools. For some tasks, computers can be a million times as powerful as man. In other of the many ways we can use computers, they will be much less than a million times as powerful as we are alone.

A COMPUTER IS:

Having reviewed outstanding characteristics of computers as they appear to users, we are now ready for a glimpse of some technical details. We have come to a fundamental question: What is a computer?

A computer is a *programmable electronic data processing system*. Computers have been called many other names, but this definition gives us a good start towards understanding computers technically. We'll consider the significant meaning of each word, except "programmable" and "system", which are discussed elsewhere.

Because a home computer is electronic, it is powered by electricity, generally from a 110 volt wall outlet, just as other electronic appliances are. Computers are solid-state electronics, so need relatively little power and are not likely to wear out soon or to need many repairs. Many parts of a computer are less likely to fail after they have been used for several months or years than when they are new. However, computers are very complex and some parts are not solid-state, so the availability of repairs should be considered when selecting a computer.

A computer works with *data*, or information. This characteristic is similar to the fact that our minds work with ideas, rather than the physical things our bodies work with. However, as previously mentioned, a computer can control almost any physical activity.

The data a computer works with must be objective and precisely expressed. Computers can't work with feelings or subjective ideas unless they are expressed objectively. Neither computer data nor computer results are ambiguous, despite the fact that their meanings aren't always obvious to all people.

All computer data is expressed in a very simple numeric code. Knowledge of this coding is necessary for the user who wants to understand the internal operations of a computer. However, a computer can translate its codes into words or numbers that you are familiar with, so computer codes need not be of concern to you. You can use today's computers without consciously learning any codes.

A computer processes data. Processing means changing, generally into a more useful form. A computer changes data it has been given into data that is more useful, or more fun, for its individual user. The processing a computer does is whatever its program directs it to do.

MICROCOMPUTERS

The type of computer generally used in homes is called a *microcomputer*. Most of the computers sold in local computer stores are microcomputers.

A microcomputer is a computer that is small in size and price. A microcomputer need not be small in power or capability. There is no exact definition that precisely distinguishes microcomputers from other types of computers, just as there is no commonly accepted rule that distinguishes a microskirt from a skirt. This paper discusses computers in general and emphasizes the characteristics of home microcomputers.

INPUT

To start a computer working you give it a program and data. Communicating these things to a computer is called *inputting*. The computer parts you use are called input devices.

You often communicate to a home computer in ways similar to some you use to communicate information and instructions to people. You can communicate to a computer by pressing keys, or buttons, on a keyboard that is very similar to that of an electric typewriter. Some lower cost computers have keyboards that are much like the keyboard of an electric calculator. All computer keys are electrical switches.

Other types of switches used for computer input are similar to the on-off light switches in your home. Computers with many switches of this type are best suited for the user who wants to know much about computers and work with them in intimate detail.

A variety of other types of input devices are used with home computers. Some computers can understand spoken words, but this capability is not yet common for home computers.

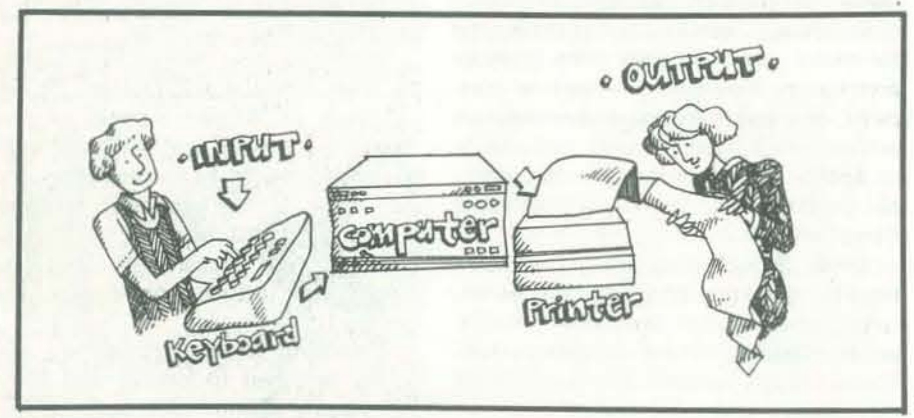
PROCESSOR

The key part of a computer is a processor. A *processor* is a group of electronic circuits which actually performs the calculating and the logic operations comprising a program. A processor performs the same functions for a computer that a brain (less its memory function) does for an animal.

The processor of a microcomputer is a microprocessor. This is a small electronic component, one kind of the wide variety of components called *chips*. A microprocessor is a small form of a processor. Microprocessors are widely used as control devices, in many kinds of machines besides microcomputers.

You may hear a microprocessor called a microcomputer or a processor called a computer. However, there are critical differences for the user.

A processor is a control element, a part of a machine that directs, by following a program, the rest of the machine to do whatever it is supposed to do. A microprocessor can be part of a calculator, part of a microwave oven, or part of a sewing machine. In none of these uses is a microprocessor a computer. Nor is the calculator or the microwave oven a computer just because it is controlled by a microprocessor.



A computer, or a microcomputer, is a complete machine, including all the parts and accessories needed to do whatever it is used for. A computer can operate entirely on its own, needing only a source of energy and, at the start, directions from a person. A computer is built so you can input your instructions to it, and so it can communicate the results of its processing to you.

MEMORY

A computer uses parts called *memory* to store, or remember, the programs and data that have been input and the results that perform the memory function are chips, similar to microprocessor chips. In many computers, memory chips are physically part of the processor.

A home computer can also remember information by storing it on a magnetic tape cassette, such as the type you use on a cassette recorder to record, and thus remember, music or words. A cassette recorder can also be an input device because you can use it to read into your computer programs that other people have written and recorded on their computers.

Another type of home computer memory device is a relatively expensive and powerful device called a floppy disc drive. This is a recording and playback device that uses a rapidly spinning disc about the size of a small phonograph record.

OUTPUT

The final common functional part of a computer is an *output* device, which a computer uses to communicate to you. The most common home computer output device is one you are quite familiar with – a TV screen, on which the computer displays words, numbers, and sometimes pictures. This screen can be that of a standard home TV, or of a specially designed device.

Computers can also display their output by using lights that form numbers and letters, as do the output lights on an electronic calculator. Computers can also use individual lights that are either off or on, and thus communicate to you the status of various processes and coded data. Computers can also communicate to you by printing on paper and, less commonly today, by "speaking" words.

INTERFACE

Many computers have several groups of electronic components, called *interfaces*, to allow the computer to communicate with devices not normally compatible with the computer. An interface is like a translator. You can, for example, buy an interface so your home computer can display data on a TV set you now have. Often some interfaces are built as part of a computer, while interfaces for less common input, output and memory devices are options.

POWER SUPPLY

Computers have one other part which the potential buyer should consider. A computer uses electrical power, but of a low voltage, similar to that used by electronic calculators and similar devices. This low voltage is supplied by a group of electronic components called a *power supply*. Probably you've used an "AC adaptor" to supply power to a device which normally uses batteries; a computer power supply has the same function as an AC adaptor.

A power supply is sometimes included as a standard part of a computer, is sometimes an extra cost item, and is sometimes not even available from some vendors. Also, a large microcomputer system with many accessories needs a more powerful supply than a small system, and perhaps more power than can be supplied by a computer as sold. Check before you buy.

PROGRAMS

A program is not usually considered a part of a computer because it doesn't exist physically, just as an idea doesn't. Programs are often called *software*, while the physically existant computer and accessories are called *hardware*. Software is essential to the use of a computer, and some types of programs are commonly purchased with the computer or as accessories. The availability of software should be a key computer selection factor for most potential users.

The most important type of program generally furnished by a computer manufacturer is a language translator. This program translates between your English language and the computer's machine language so your computer can understand the programs of instructions you communicate to it. This translator program may be called an *interpreter*, an *assembler*, or a *compiler*.

The most common English-like language used to program home computers is called BASIC. This is one of several languages, called *high level languages*, that are easy for casual programmers to use. BASIC interpreters are available for many computers. For some computers, more than one BASIC interpreter is available. Different BASIC interpreters vary in the number and the power of the instruction words they recognize and in the accessories and capabilities they require of the computers on which they operate.

Lower cost computers have no language translator programs, so you must communicate with them in machine language. Although such communication is educational for the user who wishes to understand how a computer works in detail, it isn't in an easy language for a person to use.

The other major category of home computer programs includes those that give a computer the capability to do whatever its owner wants. Some sources of programs of this type include:

1. The manufacturer and vendor of your computer
2. Other vendors
3. Program libraries, which loan or give copies of programs, at little or no charge
4. Friends and computer club acquaintances
5. Your own ideas —this is the type of program you can write yourself.

Although some programs will be easy for you to write, others could take several months of hard work. A discussion of the many factors affecting program procurement decisions and the decisions of which programs you should write yourself is beyond the scope of this article. A brief summary is that there are many ways you can get programs and many unexpected options and decision factors that may be involved. The availability of programs for doing specific jobs certainly can be a major computer selection decision factor.

USES OF HOME COMPUTERS

Having now covered the technical characteristics of computers, we are ready to consider some of the things they can do for us. Even though many people understand the immense potential computers have for helping in our homes, they still ask: What are home computers used for today? This is a reasonable question, and probably the most common one for a home computing newcomer. The answer is: Mostly for fun.

Although computers have proven that they have many other potential uses, the reason relatively few people are using computers for serious purposes is that most people can't seem to quit playing with their computer. I see nothing wrong with that. Work, for many people, should be done at work. Those who can spend most of their time at home for enjoyment seem to have achieved an important measure of life's success.

The most common type of home computer owner today is the builder, a craftsman or craftswoman. For these many people, understanding, building, or otherwise developing computers is an end in itself. These computer hobbyists are similar to the painter or woodworker whose enjoyment comes from creating, rather than using or possessing.

Computers are the world's most pliable media – you can make more things of a computer than you can make of clay, paint, or wood. Creating with computers can be extremely challenging, and success extremely rewarding, because the potential for creativity, mental and physical, is almost infinite.

Other people enjoy using computers for playing games. When used for games, computers are often somewhat like TV games, today's hottest consumer item. Unlike today's common games, which are very limited and rather quickly become boring, the variety of computer games can be almost infinite.

People with home computers can keep getting new games, sometimes for less than $1 each. Better yet, they can design and use games they create themselves. And these aren't just bouncing ball games. Home computer owners program and use their computers to play cards, space war, chess, and many other games. The really fun games are ones you never heard of because they can only be played with the help of the immense power of a computer.

But computers can be very practically useful, as proven by their use in, and great benefits to, almost all of commerce, industry, and medicine. Easily affordable home computers can be at least as powerful as many of those that are doing thousands of different jobs in business today.

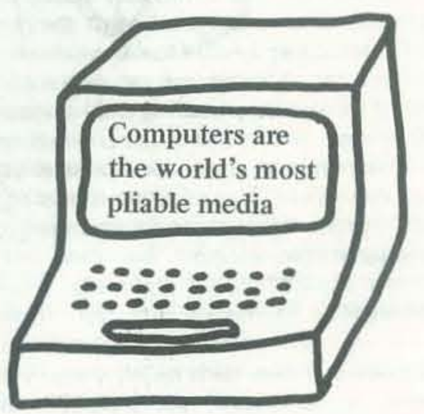
Families use computers in home offices to keep records and to handle accounting chores. Some also plan stock and commodity futures buying and selling. Home businessmen use computers in many occupations.

Computers control heat and other environmental factors so an entire building, such as a home, is more comfortable and so energy is used more efficiently. Home security is another service, meeting an unfortunately increasing need, that computers are being developed to help supply.

Education is another of today's uses of home computers. As educational tools, computers are helping to teach subjects ranging from first grade addition to financial and engineering math, and a wide range of other subjects. Also, many people are using computers to teach themselves about computing, a field with considerable employment potential. In addition to vocational benefits, learning about computers is both enjoyable and practical because of the rapidly increasing role computers are coming to play in the everyday lives of each of us.

Many computer owners have talked of the great potential of computers for keeping track of kitchen supplies, planning menus and recipes, and for many similar household chores. For some reason, these uses seem to be like most household chores, they end up in a "tomorrow" category, even though their long range benefit can include considerable work savings.

Returning to fun applications, computers are used to catalog and control the playing of musical recordings. Computers are also used to make music. There



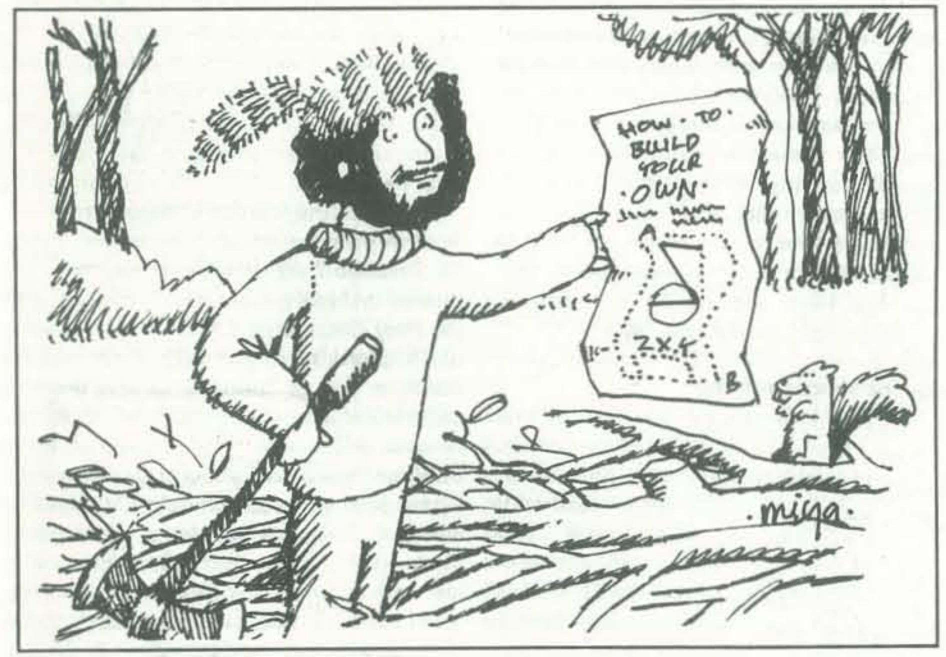
is even a magazine devoted to the topic of computer music.

Electronic hobbies are another home computer application. When electric train layouts are so complex that parent and child together can't control them, computers keep the home railroad running smoothly. Some amateur radio hams make extensive use of computers in their hobby and have generally been the pioneers in home computing.

The list of what computers can do, of what you can do with your computer, can go on and on. Aside from hopefully giving you some more reasons for getting involved in computing, the preceding examples of computer uses should further support the idea that a computer can do almost anything.

HOME COMPUTERS TODAY

But a computer in your home? Lots of people have computers in their homes already. But keep in mind that these people are the pioneers. They didn't have to cut down logs with an axe to build their own cabin. But some of them have done as much pioneering in other ways. They have done the hard work. You can buy a prefabricated, ready built computer, ready to move in. However, there is plenty of wilderness left in computing. If you want to be a pioneer, there are many op-



portunities for you to go ahead into the unknown and develop techniques and products. And there are lots of people who will follow and much appreciate any real discoveries or progress you make.

This is one of the choices you can make as you look at today's computers, and as you decide how you will become involved in computing. You can buy the equivalent of 40 acres of wilderness, or a prefab in a subdivision, or many things in between. You can totally design and build your own computer. Or you can get one that you can plug in and have start immediately to work for you. There are many available options between these extremes. Your home computer can be what you want it to be.

ELEMENTARY COMPUTERS

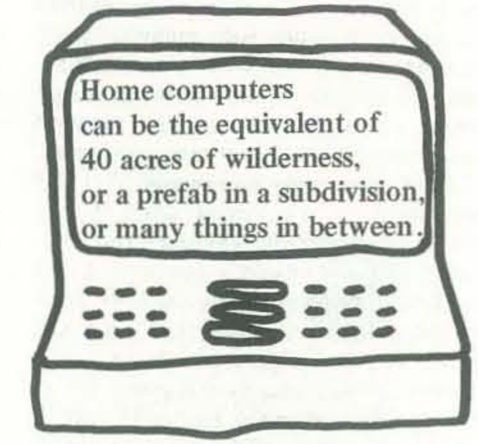
You can get a very basic type of computer. This will be relatively inexpensive, costing between $100 and $250, generally less power supply. It won't have a fancy cover, and probably will have no cover at all. All the parts of this type of computer may be on one piece of plastic-like material called a printed circuit board, or just "board" or "card".

You will have to do lots of detailed work to program and use such a computer, especially because programming generally must be done in machine language. The power of this type of computer is limited – you can't easily add many accessories, nor can you easily connect such a computer to other devices to control them.

This type of computer is a good training device, an educational tool. The lack of fancy accessories means that you work directly with your computer, with little interference. Therefore, you can learn, by doing, much about the details of how a computer works.

EASY-TO-USE COMPUTERS

Some people don't want to work closely with a computer. They would rather have their computer follow their general instructions. Such people aren't especially interested in knowing the details of how a computer works. They would rather learn how to use a computer to solve a particular type of problem, control a particular process, or help them in a field un-



related to computing.

Computers designed for this type of user are available. One key characteristic of this type of computer is that you program it by using a high level language. As we discussed earlier, a high level language is easy for people to use.

These computers are generally built on several printed circuit boards which plug into guides and connectors mounted on the computer frame. Computer capabilities are changed by plugging new boards into appropriate spaces, as long as there is room available. For some computers of this type, especially "Altair bus compatible" or "S-100 bus" computers, many accessories are available that can be added. For other types of computers, only a few accessories are available. Consider what you may need and want.

Prices of computers in this category generally start at about $500. A system with enough power and accessories to use a high level language may cost $1000. The price of a full system, such as might be used for a business application, can easily reach $5000.

KITS?

Another primary choice when selecting your home computer is whether to purchase a kit of parts, or an already assembled computer. A kit, like the traditional model airplane, requires your work and time to put it together. A computer kit often takes from 50 to 200 hours to assemble; accessory kits generally require less time. Time requirements vary depending on your skill, the ease of assembly of the kit, and your luck in not encountering problems which can take many hours to diagnose and repair.

The major benefit of a kit is that it saves money. An assembled computing product generally costs from 30% to 50% more than a kit. Another benefit of a kit is the experience that assembly gives. However, the person who has so little experience that he would benefit probably would encounter considerable problem in assembling a working computer.

Most computing kits require skill and knowledge in electronics work, sometimes a considerable amount. The decision whether to build a computer kit is not one to be made lightly. Anyone, after he has enough skill, equipment, time, and expert help available, can build a kit. The amount of these resources required for assembly and successful use varies widely among kits. The quality of instructions provided also varies considerably. These are all factors to consider when selecting your computer.

HELP!

Although good instructions and the availability of expert help are essential to the novice kit builder, these aids are important to every computer user. Computers today are too complex to be understood completely by most people. Because computers have so much potential, appropriate understanding is very important for using a computer well. Because skill requirements vary widely among types of computers, a solution to the need for skill is to choose a computer which needs the amount and types of skill you have or plan to have.

A special resource to consider when evaluating sources of computer skills is the dealer from whom you buy your computer. This is the person from whom you will also probably get the several types of support you will need to use your computer and to keep it operational. A dealer who is knowledgeable about the equipment he sells, and is willing to share this knowledge, can be a great help in using a computer. So the decision of whether to buy from a dealer, and the selection of your dealer are both important home computing decisions.

CONCLUSION

After this quick overview, you probably are much more confused than when you started because you have learned a little and can better see how much there is to learn. At least you have some understanding of the fact that much information about microcomputers is new, even to the most knowledgeable and experienced persons. The knowledge you now have is greater than that of many other people. So go ahead: talk to owners of home computers and visit computer exhibits and stores. Most of all, have fun as you plan and learn about your new helper, your own home computer. ■

WOMEN & COMPUTERS:

A Dialogue

Dear People,

Consider this a letter of desperation! I am the part owner of a brand new Poly 88 Systems 7 along with ill-assorted enthusiastic family members. We bought the Poly because of the recommendation of *Dr. Dobb's* . . . but aside from understanding that he thought it was good I understood absolutely nothing of what he said and therein lies the rub—everything written in every magazine and book I have seen assumes a technical and mathematical background which I simply don't have. It has been 25 years since I last looked at an algebra book and my math teachers then passed me rather than have to deal with my incredible inability to understand anything beyond add, subtract, multiply and divide. As a science fiction nut I dreamt of becoming a nuclear physicist but my high school Physics teacher, a quiet and dignified ex-Colonel, looked at me one day in anguish and asked hopefully whether I was going to make this my last science course. Why break a man's heart—right? So no more physical sciences. About five years ago, I began college and now have an M.A. in anthro (I failed as a housewife) and am dabbling at the edges of a Ph.D. Clearly, I am not totally beyond redemption.

Now, being a sci-fi nut and having a 16-year old son who is computer crazy and a gentleman friend equally enthused we all got this machine of the future and I am determined to learn to use it. I decided to appeal to you folks, since given your name, I thought you might have some commitment to really making computer use available to *people,* not just technologically or mathematically oriented people but plain dodos like me. Felsentein's page in your November-December issue on "What is a Chip" was the first thing I read which was totally comprehensible, so it *is* possible to explain computers simply.

We bought 16K BASIC with our Poly and after much trial and error learned how to load it despite the manual. I find the entire Poly manual beyond comprehension. As a matter of fact, I thought some of the sentences might be magical incantations but so far no lead into gold or anything so apparently it is English. But it is not really useful for me. I bought a copy of *How to Buy and Use Minicomputers & Microcomputers* but by the time he got to explaining hex (vaguely understood) I felt overwhelmed, besides which nobody ever says, "This you must understand to program and this is extra." I drive a car quite well without understanding the theory of internal combustion; what is essential for me to know to become an adequate programmer and user and what is extra?

Every group develops its own ingroup and outgroup delimiters and one way of course is to develop a language which only the insiders can understand. This excludes those who can't speak the language and gives the in-group a feeling of solidarity.

All of us do it—anthropologists included. But isn't it time computer people look around and let some of us who will never learn the wonderful complex terminology in? How about really making computers available to the people regardless of race, creed, sex or technological background?

This brings me to some comments about women and computers. The computer world tends to be a man's world. Without going into a long thing on how women are socialized away from math, technology, etc., the truth is many of us women lack the life experiences of males in dealing with electricity, machine building and repair and thus find entering the computer world much more difficult. At the Computer Club meeting in Detroit that my friend went to, there were about 100 men and three or four women. Why? What about the wives and lady friends of all those males? Is it that they are not interested or that they have been lead to believe that personal computing is only for mechanical whizzes? I would hazard a guess that most women would lose interest because the hobby is aimed at male skills. But it shouldn't be that way and somebody has to start changing things. *People's Computers* seems like the kind of outfit that could do that, that could in a non-sexist and non-elitist way (don't talk down to us; it ain't IQ that's lacking, it's background) provide an introduction to computers for all those (and there are men too who could benefit) who have been put off by the run of the mill info on computers. How about Felsenstein taking on the job? I would be delighted to make suggestions as to what could be included, proofread or anything to help.

Please, somebody—help!

Yours for computers for *all* the people,

Annette Ran
17250 Cornell
Southfield MI 48075

P.S. Upon reading this letter people may react negatively, wondering why we bought a computer in the first place, and thinking that if we lacked the scientific and technical background we should have stayed out of the computer world. So, perhaps some explanation is in order.

First of all, why did we buy our lovely Poly 88? Both Jim (my fiance) and I have been avid science fiction readers since our early teens. The idea that someday everyone would have a personal computer (see for example, Arthur C. Clarke's *Imperial Earth*) was for us part of what future technology would surely bring. When we first learned that home computers were available *now,* it was like a piece of the future we had dreamed about come into the present. We assumed that the computer would have many practical applications but still our primary reason for purchasing it was because it was a step into the future. It's fun; it's exciting; it's challenging.

To respond to those who want to keep home computers the sole property of those who speak the jargon and wield the soldering irons I want to say this. Whether you like it or not computers are going to have to enter the everyday world if only because the economics of manufacture dictate the need for an expanded market. Home computers can revolutionize energy conservation, education, household management, etc., etc. Those of you who have developed the expertise in the computer field can either be a forefront for teaching the rest of us or you can sit in your exclusive groups and grumble about the good old days when only the deserving few had home computers. One way or another, however, with you or without you, the home computer industry is going to have to enter the everyday world. People in the industry are going to have to speak in a language we can all understand. It is, in a way, a challange to those of you who have been the pioneers. We novices need your experience. How about it?

Dear Annette Ran:

I have known many women successful in the computing field from assembly-line workers through directors of college computer science departments. As a fairly recent area of employment, many of the previous sex obstacles have been overcome. The problem of being a beginner in a highly technological field is greater than that of being male or female.

Regardless of what anyone says, a computer is a highly complex piece of electronic equipment. Since most computer manuals are written by technically oriented people for technically oriented readers, they are difficult for a beginner to understand. I agree with you that manuals are not easily understood and *this must change.* Manufacturers have failed to realize the varied background of their consumers. At the same time, I have seen 10-14 year old children in some of my classes using computers after 15 minutes of instruction and programming them within an hour. Therefore, the "male skills" you mention are *not necessary* for computer use. These students had no "life experiences of males in dealing with electricity. . ."

The state of mind you must reach is not that you are a *FEMALE,* but that you are a *beginner.* As a beginning writer of materials for computer beginners, I find that it is a tremendous challenge to write about a complex machine, such as a computer, in simple, easy to understand language.

It is not that you are a FEMALE, but that you are a BEGINNER.

There are many levels of computer usage. 1) At the machine language level, you are working very closely with the computer itself. You converse with it in its own language. The Data Handler articles in *People's Computers* are aimed at this level. 2) Assembly language is the next step up. At this level, you work with *MNEMONIC* codes which are translated by an "assembler" into machine language for action. 3) Higher level languages such as BASIC tend to be more like English, and are more readily usable by the beginner.

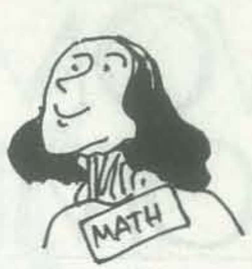
The computer itself is technical. Programming is a matter of logically putting down the commands the computer must perform. It is a matter of learning a language.

Before anyone buys a computer, they should decide what they are going to use it for - recreation, education, controlling external devices, etc.

From your letter I think you fall into the recreation-education category. Dymax's new magazine *Calculators/Computers* should prove useful if this is true. Two books published by Dymax may also be helpful to you: *Your Home Computer* by James S. White, and *Instant BASIC* by Jerald R. Brown. Each costs $6 through the Bookstore of People's Computer Company.

Whatever happens - don't give up. Keep reading *People's Computers* - we'll try to help.

Don Inman, Editor
Calculators/Computers
Dymax
Box 310
Menlo Park CA 94025



Dear Annette:

There seem to be two separate issues here -- the problems of an inexperienced user getting started on a badly-documented system and the shortage of women in computing.

I'm not involved in hobby computing, but from what I hear from those who are, the machines now available are unreliable, poorly documented, and come with very little software. They are just not a good place to start, and probably won't be for a few years. If you already have one, though, that isn't much help. I think it would be worthwhile to step back and pick up some background by learning how to use a commercial computer. A lot of community colleges, etc. have programming courses, or you can use your school's computer and teach yourself. It's *much* easier to learn how to program when you have good software and don't have to worry about machine errors. The documentation for these systems is usually pretty good at telling you what you need to know and ignoring the rest. Also, there are lots of good introductory programming books that don't require much math.

The other thing that would help you get started is to find someone with experience on your machine (or another) who is willing to spend some time with you. You might find such a person from the people who sold you your machine, in a local hobby club, or maybe even through People's Computer Company. Even a small amount of help would save you from a lot of wasted effort.

Now, about women in computing. There are more women in this field than in other areas of science and engineering, but there still aren't very many of us. I think you are right about women being deterred by the male image that technical subjects have, and by their own lack of experience and confidence. For the future, it's important to encourage girls to feel comfortable with math, and that may involve changing the attitudes, or at least the behavior, of a lot of teachers -- male and female both. People are working on this, but more is needed.

Even with bad early experiences it is possible to catch up. Mills College in Oakland, for example, has a very good program for women with weak backgrounds in math. I think that anyone who wants to do serious work with computers will have to grit their teeth and learn the math -- it's important. But a deep technical background isn't necessary for people who want to use computers casually. This is recognized in the commercial computer market, and there is a lot of good documentation available here (as well as some terrible stuff). The manufacturers of hobby machines should be encouraged to go and do likewise. In their defense, it isn't easy to write good introductory material -- you probably have to hire new people to do it. The idea of articles for the non-specialist in *People's Computers* is great. No reason to aim it particularly at women -- just the beginners. These articles would also be useful for introducing experts in one area to new fields.

There are more women in the computing field than in other areas of science and engineering, but there still aren't very many of us.

Susan Owicki
Digital Systems Laboratory
Stanford University
Stanford CA 94305

Dear Annette:

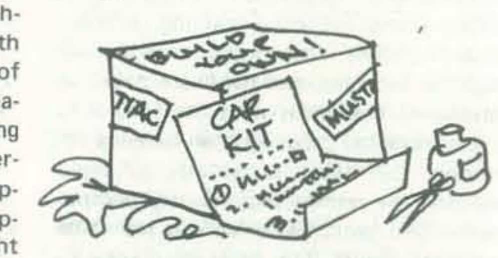
I was delighted to read your letter decrying the lack of non-technical material available for the unsophisticated microcomputer user. I empathize with your position because I am actively involved with microcomputers and microcomputer technology and yet my technical background is limited to theoretical mathematics and social sciences. Now you might think that theoretical mathematics is a big step in the right direction -- and I won't deny that it helps -- but I'm still a long way from understanding integrated circuits, semiconductors, and how to use a soldering iron or voltmeter. What I have learned about microcomputer technology comes mostly from patient explanations by electronic engineers.

I might point out that there are two distinct classes of microcomputer users -- even among technically-oriented people. These catagories are hardware people and software people, with very little apparent overlap between the two. My own perception of hardware people is that they genuinely love the equipment - not so much what one does with it, but the actual technology and design. They are as enthusiastic about well-designed memory boards and state-of-the-art technology as I am about Beethoven's Ninth Symphony. The design is often a thing of beauty in itself. Software people, alternatively, conform more to Annette's driving analogy. They don't care about the internal configuration of computer equipment. They are more concerned with applications. And these applications might involve instrument control, instructional uses, business systems, or turning on your lawn sprinklers.

I happen to fall into the latter class. I don't really care about circuit loops and shielded motherboards. But I do care about good software that will help me do text-editing, business statistics, and a wide range of what I consider to be useful applications.

Regardless of which class one falls into, however, there is still almost no literature available that provides an understandable, introductory explanation of either computer hardware or software. At this moment in time, the microcomputer industry is aimed at the hardware half of the user spectrum. And more than that, the literature is aimed at the technically-sophisticated hardware person. However, I should say, in defense of the industry, that this is not an unreasonable approach. An understanding of computer hardware does require some technical background. And it is unreasonable to assume that hardware manufacturers, trade journals, or computer stores should provide non-technical persons with an introductory course in electronics. None of us expect RCA, Sony, or Zenith to supply us with documentation explaining the fundamental operation of our television sets. So anyone with a serious interest in computer hardware should get some kind of technical training -- preferably, an introductory course in electronics.

The software user faces slightly different problems. I don't have to understand how my stereo equipment is designed and constructed in order to turn it on, adjust the volume, run a turntable, play a record, and enjoy the music. I foresee home computers eventually providing the same kind of service. The home computer will be no more mysterious (or less mysterious, for that matter) than the telephone, the television, or the stereo. Most of us own and efficiently use all of those things without understanding the internal workings. And most of us would no more think of buying a television set in kit form that we would of buying a car in kit form.



It sounds like the reason you are experiencing so much difficulty is that you are in the position of wanting to "drive your car" without wanting to become a mechanic. The microcomputer business is just developing to the point where the hardware required for a wide range of applications is now available at a reasonable cost. Unfortunately, the development of software to support applications is lagging far behind hardware development. I won't go into all the reasons why software is so slow in coming, but it puts us non-hardware people in a difficult position. We are all faced with the same kinds of problems shared by the first purchasers of television sets. We own or would like to own these marvelous, mysterious, high technology devices but we don't have any programming of a quality comparable to the hardware. We're stuck with a television set for which there does not yet exist Monday

Night Football in living color. All of that will come, but it takes time.

I also empathize with your comments on women and computers. Sure computer clubs are about 99% men. But that is really not surprising. More men happen to be in technical fields. If you look at the number of women in mathematics, engineering and computer science, it isn't difficult to understand why so few women are currently involved with home computers. This situation will certainly change as more and more women enter technical fields. I can assure you that the problems of dealing with computer hardware and computer terminology are not restricted to women -- they are problems that all non-technical people experience. Not very comforting though, is it.

Until microcomputer applications are adapted to the consumer market, there is no easy solution to the problems concerning lack of introductory material for novice users at either the hardware or software level. So perhaps the first step is simply to encourage the production of such materials. However, I re-emphasize that anyone with a serious interest in understanding computer hardware should obtain some technical training. I don't mean to deny that much introductory material for computer hardware could be developed that deals with the subject in a less technical capacity than existing literature. But there is no way to avoid learning the terminology, system components, and technical details in the same way you would have to it you were assembling a car, studying medicine, or practicing law.

Applications-oriented users, however, do not require or need this kind of technical background. They simply need some basic, non-technical skills that will allow them to use the hardware. Unfortunately, literature explaining what these skills are doesn't really exist -- mostly because you can't develop software without the hardware and the hardware has only existed for a little more than a year.

There are a number of solutions. I will briefly suggest a few. First, some introductory materials could be obtained if we demand of journal editors that they solicit articles aimed at the novice user. This is an admittedly slow process and not too many technical writers can write for non-technical people. But it would be a start. We could organize a national group or association that would promote instructional and introductory materials for microcomputers which are aimed at non-technical users. We might approach the National Science Foundation to obtain funding for the development of workshops aimed at novice users, for the preparation of introductory courses to be held at adult education classes, or junior colleges, or for the preparation of introductory courses to be televised on instructional television stations. We might set up a clearinghouse for journals, articles, books and other materials aimed at the home computer enthusiast. I am sure that some solution of this type would attract a lot of women and men, who (like you) would be glad to be included, to help, to proofread, and to provide differing degrees of expertise and experience which would benefit us all. I do think that we need a continuing dialogue on the problems facing non-technical microcomputer users and on possible strategies for dealing with the problem. So let's find a way to keep it up.

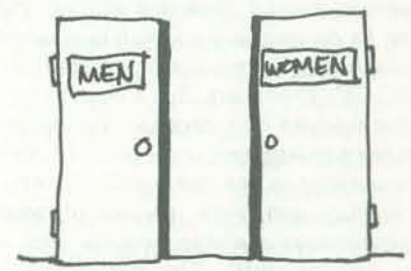
Alice E. Ahlgren, Ph.D.
Marketing Manager
CROMEMCO, INC.

Many thanks to you, Annette Ran, for taking the time to remind those at *People's Computers* that the computer world has been male-dominated for far too long, and that the time for taking steps toward changing this is now! As I read your letter some additional questions came to mind, and I would like to share my feelings and reactions with you and other interested readers. I believe that we computerists need to seriously consider why so few women are in our ranks, and make an effort to help women feel that they are welcome to join us.

The computer world has been male-dominated for far too long.

THE PROBLEM

(1) Why do so many men and women believe that computers (and math and science) are male domains?
(2) How do these beliefs affect those women who are interested in learning these subjects and who try to enter these fields?
(3) What can be done to stop perpetuation of the myth that women shouldn't and/or can't succeed in these areas, and to increase the numbers of women who deal confidently and competently with the quantitative and technological aspects of our society?



ARE SCIENCE AND TECHNOLOGY MALE DOMAINS?

The first question is, perhaps, the easiest to answer. Just take a look around you. How many female engineers or physicists or astronomers or chemists do you know? How many have you ever seen or even heard about? Of how many women scientists who have contributed significantly in their fields are you aware? (There have been many besides Madame Curie.)

Walk into a participatory science center, such as the Lawrence Hall of Science in Berkeley or the Exploratorium in San Francisco. Count the numbers of males and females you see; guess what you'll find? Look even more closely and observe the behavior of those who have come. Among the females, how many of them are actually participating, investigating and exploring, as opposed to passively looking on? Unfortunately my observations indicate that even among those outnumbered females who do come to these centers, unlike most of the male visitors, many do not get very involved.

Think back to your school years. How many of the math and science teachers were female? When you needed help with math homework, to whom did you go, to your mother or father? In textbooks and in nearly all media presentations, in how many different roles were women ever depicted besides as mother, secretary, nurse or librarian? Whenever you think of "a scientist", what picture comes to mind?

Look back even further to the preschool years. What kinds of toys are given to girls and what do boys receive? Answer: girls play with dolls and dresses and learn early that female sucess is a function of pleasing males, not of independence nor self-reliance. Boys, on the other hand, get models to build, learning early to explore, to question, to experiment with the way things work, to try and figure things out. Boys are encouraged to investigate and solve problems, to take things apart and reconstruct. Girls are taught to depend on males rather than to do these things for themselves.

BUT SOME WOMEN ENTER THESE "MALE" FIELDS

Considering the environmental messages girls receive about appropriate roles for women, and the dearth of female scientists as role models, it is not surprising that most young women do not even consider aspiring to enter technical fields. Since most girls lack these aspirations *and* the early experiences which facilitate successful participation, it is not hard to understand why most girls don't even bother to take math and science when it becomes optional in high school, thus further increasing the gap between male and female preparation. Considering this gap in preparation, it is no wonder that many people believe boys are better than girls at math or women just aren't good with mechanics or electronics. However, a key point which many people either fail to realize or choose to ignore was well-put by Annette: "It ain't IQ that's lacking. It's background." Many females simply lack the prerequisite experiences for dealing adequately in technical areas.

Some girls, fortunately, do develop excellent "male" skills; many continue on to enter scientific and technological fields. How do these women make it? Often they have received extra support which most women lack. Some parents are enlightened enough to provide similar experiences for sons and daughters. Some teachers do encourage women with high mathmatical or scientific potential to go on, though many still fail to encourage or even discourage females. Some counselors do recommend that a talented yourg woman consider becoming an engineer, though few even think to mention this possibility. And some women lacking any outside support, succeed despite external pressure, due to their own inner strength and determination by daring to be different and being good at it.

ACTION PROGRAMS

All females who show potential should receive appropriate support and encouragement. It is extremely important that as scientists, technologists, and educators we realize that girls are not receiving the experiences necessary for developing competence, confidence or even literacy in our fields. We can and must get involved in programs to reverse this trend. I would like to direct the reader's attention to the article entitled "Women and Math Projects: Lawrence Hall of Science," describing a series of programs of this type which are already underway. These programs all present opportunities for young women which offer:

a) *role models*– women who have succeeded in traditionally male fields, who are willing to teach young women what they've learned, to discuss their own experiences and to show that these fields are not only accessible to women, but challenging and rewarding options for women to consider
b) *a supportive environment*– an attitude of understanding the reasons why women/girls may hesitate to get involved, a spirit of encouragement, and a sincere belief that women *can* succeed if they elect to face the challenges
c) *validating experiences*– a chance to actually experiment and gain realistic, yet successful experiences working with the tools of the trade
d) *continuing support*– names of women and men who are willing to be contacted even after these events, for discussions about career options and for career guidance– a 'buddy' system.

In the San Francisco Bay Area programs of this type are being offered on an on-going basis. Anyone out there want to participate? Let me know. Perhaps readers in other areas will consider setting up similar programs. *People's Computers* will be glad to publicize these events.

COMPUTER CONFERENCE FOR WOMEN

I would also like to propose that we set up another conference, based on the model just outlined, but focused specifically on educating girls and women about computers. What do you think? Workshops might include anything from "Programming in BASIC" to "Designing circuits" and "Building a microcomputer." Computerists from People's Computer Company, Lawrence Hall of Science, Cromemco and Mills College have already begun discussing the possibility of sponsoring such a local event. We would love to hear from anyone interested in helping with plans, financing, donation of equipment, etc. Do you want to get involved?

Thanks again, Annette, for motivating this discussion, and thanks to Phyllis Cole and *People's Computers* for providing the appropriate format.

Rita Liff
Visiting Assistant Professor
Department of Mathematics and Computer Science
Mills College
Oakland, CA 94613

The DOT and the Line

BY DEBBIE SLACK HUGHES AND PHYLLIS COLE

In the not too distant future, computer graphics will enable you and your computer to explore all sorts of new fields. Would-be artists will have a whole new medium to explore: 'painting' on color-TV screens with 'paints' and 'brushes' they design themselves. Computer programs can be written to allow do-it-yourself architects some sophisticated assistance in designing and redesigning floor plans. Dancers can do computer-assisted choreography, cartographers get help in map making, and stargazers can study the stars via 'annotated' star maps.

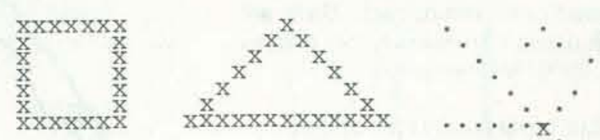
Using computers to make pictures is not new, but computer graphics as a tool for small and home computers is still a rapidly expanding field. Now portable and (more or less) affordable computers offer interesting applications. Several companies market systems integrating microprocessors with color TVs, and the number of interesting programs available for such systems is steadily increasing.

Just how can computers be used to draw pictures? The object of this article is to provide a brief, non-technical introduction to one approach to computer graphics, plotting. The idea behind plotting is in some sense to trick the beholder. Imagine a row of dots: . . . .
As you space the dots closer and closer together, you eventually have what the eye perceives as part of a line, instead of a collection of dots. The Sunday funnies provide a familiar example of how this phenomenon works. With dots sufficiently close together your eye perceives a continuous spread of color, not individual dots. Just how convincing and realistic our pictures look will depend to a large extent on the size and closeness of our dots.

TYPEWRITER-LIKE PICTURES

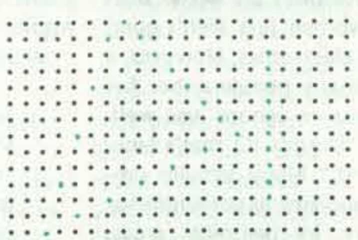
For many years an inexpensive, widely-used way of communicating with a computer has been the teletypewriter. Such devices are rather like electric typewriters and usually have a fairly standard keyboard. You can type messages to the computer on such a machine and the computer can in turn print characters onto paper. Believe it or not, many people have created lots of 'pictures' using these typewriter-like machines. Real enthusiasts manage to get recognizable Snoopies, Santa Clauses, what have you.

You can play around with a typewriter and generate such pictures; to get the computer to print such stuff on a teletypewriter is very similar. Such pictures involve some obvious difficulties: borders are not sharp, contrast is limited, it's pretty much impossible to indicate lines that are not horizontal or vertical, and you may as well forget about smooth curves. Usually, only those with no other options use teletypewriters to produce computer graphics. Here are some examples:



TV SCREEN PICTURES

Think of a TV screen as speckled with dots; pictures are made by 'turning on' dots. If you 'turn on' lots of tiny dots that are close to each other you give the eye the impression of looking at a very light area; by turning on fewer dots that are somewhat further apart you can create the impression of varying shades of gray. Just as with a teletypewriter, the basic concern is just which dots to select; but the difference with a TV screen is that there are lots and lots of dots to choose from. Just how many dots are available on a screen determine the quality of the resulting picture. Some TV screens are still not suitable for plotting smooth lines and curves, but even low-grade screens such as those used in commercial testing equipment offer better chances of success than does a teletypewriter.

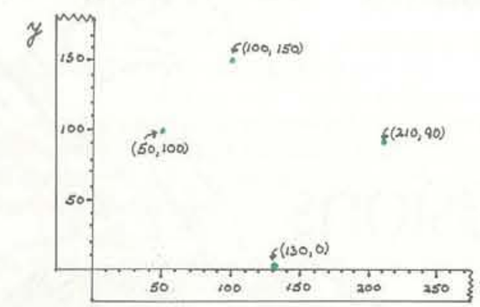


There are two methods for plotting, absolute coordinate plotting and relative coordinate plotting.

ABSOLUTE COORDINATE PLOTTING

One way to locate points on a screen is to effectively lay down two rulers and tell where the point is in relationship to these rulers. This method of identifying points is called absolute coordinate plotting. Note that two numbers are required to locate each point. The first coordinate named is the number closest to the point on the ruler marked 'x'. The second coordinate is the number closest to the point on the 'y' ruler.

Here's an example of how it works:

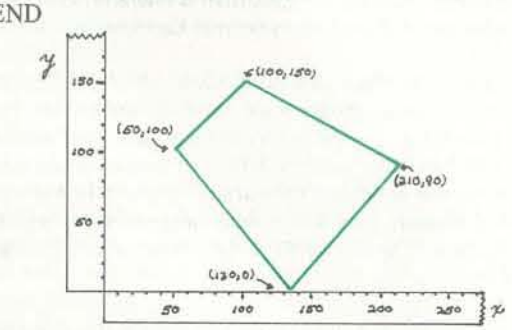


We can use absolute coordinate plotting to tell the computer which dots to 'turn on'. For example, a program to instruct the computer to illuminate the points described above might look something like this:

```
10 MARK (50,100)
20 MARK (100,150)
30 MARK (130,0)
40 MARK (210,90)
50 END
```

How can we connect the points? Do we have to specifically name the points between the points we wish to connect? Not if we're practical enough to make sure the computer language we're using has the capability of doing graphics. Such a language will allow us to do more than just illuminate the points we name. It will also contain special instructions to facilitate drawing. For example, Let's assume our computer language has a command called CONNECT. We can say CONNECT (100,150) (210,90) and then the computer will automatically figure out what dots on the screen need to be illuminated to create the impression of an apparently continuous line segment between the point (100,150) and the point (210,90). Here's a sample program and the resulting figure:

```
10 CONNECT (50,100) (100,150)
20 CONNECT (100,150) (210,90)
30 CONNECT (210,90) (130,0)
40 CONNECT (130,0) (50,100)
50 END
```



This approach offers lots more flexibility than deciding how to arrange letters or other characters on a page typed by a teletypewriter. And often a rule to calculate coordinates can be written into your program to automatically plot the points you desire.

RELATIVE COORDINATE PLOTTING

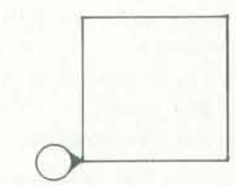
Tired of those coordinates? Here's an approach that avoids them: it's called 'relative coordinate plotting', because the computer needs to know only where you are now and where you want to go.

Imagine you have a huge sheet of paper on the floor. On the paper is a mechanical 'turtle' (see photo page 55), so called because of its shape. Your turtle is mounted on wheels and it can perform commands you send it via the computer. Your turtle is a potential artist, for it carries a pen, and can be instructed to draw with it. Some possible commands are PEN UP (to make it pick up the pen so it no longer touches the paper), PEN DOWN (to do the obvious), AHEAD N (which makes the turtle move ahead N 'steps'), and BACK N (which makes the turtle back up N 'steps'). We'll also want the turtle to be able to turn; for convenience let's have two turn commands: LEFT N, to turn left N degrees, and RIGHT N, to make it turn right N degrees. Sound like fun? Such turtles are commercially available, and have been successfully used with enthusiastic children of all ages as an introduction to computer graphics in a number of computer languages. The language LOGO, developed by Seymour Papert at MIT, is perhaps the best known. Of course you don't really need to have a robot-like turtle to instruct; the turtle can be a graphic on your TV screen.

Let's see what a turtle program might look like. Here's our turtle, waiting for instructions.

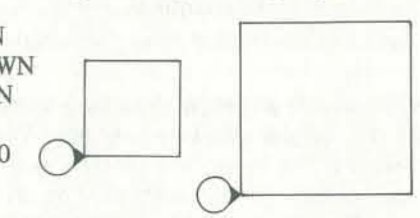
Here's the program to make him (her?) draw a square 60 units on each side. Notice that line 30 sends you back to repeat lines 10 and 20: the turtle will keep going in a square until someone stops the program.

```
10 PEN DOWN
20 AHEAD 60
30 LEFT 90
40 GO TO 20
50 END
```



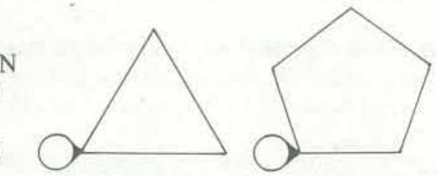
We can make a much more general program by using a command that lets us change the distance the turtle travels; when the computer sees the statement 'INPUT N' it will stop and wait for us to type in a value for N. If we choose a small N we get a small square; a large N will give a large square.

```
 5 INPUT N
10 PEN DOWN
20 AHEAD N
30 LEFT 90
40 GO TO 20
50 END
```



If we redo the program so that we INPUT the angle at which our turtle turns we have an even more general program.

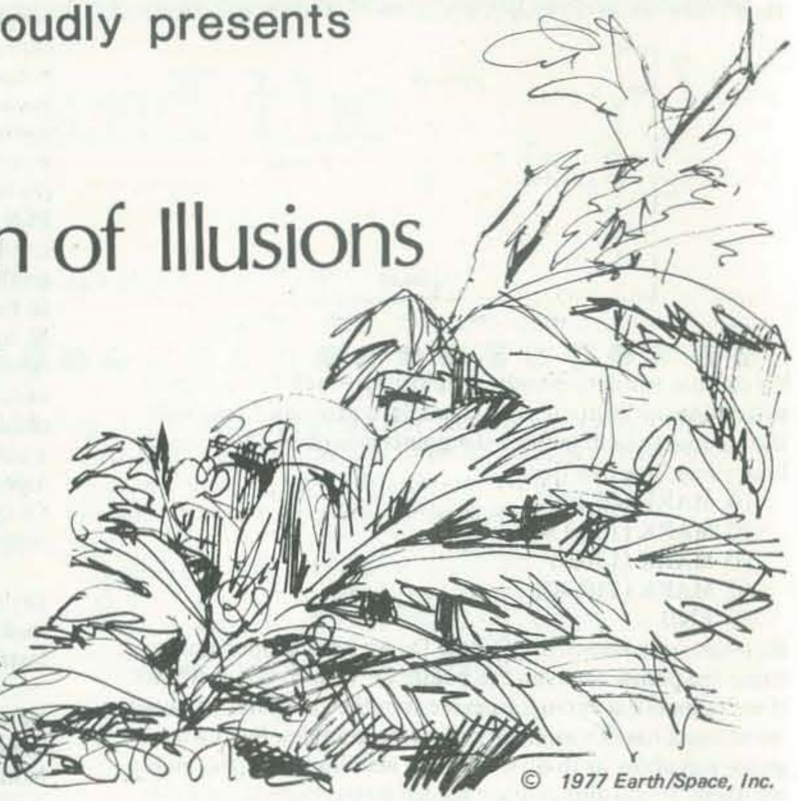
```
 5 INPUT N
 7 INPUT A
10 PEN DOWN
20 AHEAD N
30 LEFT A
40 GO TO 20
50 END
```



And it all starts with a dot . . . put enough of them together and who knows what will happen next!

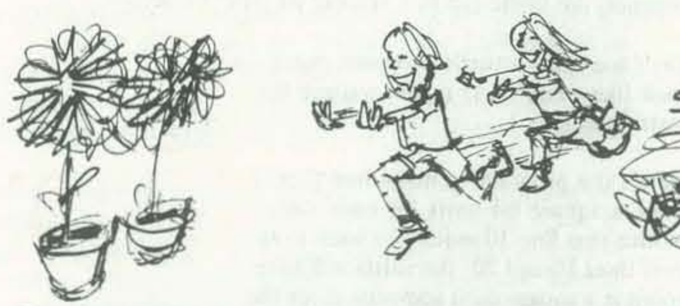
Don Quixote Starship proudly presents

SPACE COLONY: Living In A Garden of Illusions



BY MARJORIE L. STUART

ILLUSTRATIONS BY DON WOOD



© 1977 Earth/Space, Inc.

Don Quixote Starship's purpose is to facilitate participation in our advent into outer space, and to foster humane, diversified, and creative approaches to that process. Activities of DQS include expanding skills needed for space exploration through computer games and simulations which stress cooperation, communication, and dissemination of information.

Space will present us with the possibility of exploring countless alien ecologies (including space habitats), many of which will doubtless include intelligent life forms. How can we prepare ourselves for such adventures? Let us fantasize, simulate, and create as best we can such worlds here and now. We need to expose ourselves to varieties of logic, language, universal and local laws, social groupings, and methods of communication. Let's explore from as many vantage points as possible.

Here DQS presents an article discussing ways to esthetically minimize the cultural shock of long term living in a space habitat. What a fine basis for a game! Resources of such a game could include the elements used by artists and stage illusionists. The object could be to create an environment to fulfill certain criteria, such as 'house this many people in an area of this size and shape while providing an illusion that the space is thus-and-such size and shape.'

This article is reprinted with permission from Earth/Space News, *a publication promoting free enterprise in space. Earth/Space is at 4151 Middlefield, Palo Alto, CA 94303.*

Join us! Your suggestions for DQS may be sent to People's Computers *or to our alien correspondent*

Doug Seeley
833 W. 18th Ave.
Vancouver, BC, Canada V52 IW2

The living conditions depicted upon the Starship Enterprise in fiction would be esthetically oppressive in reality. However, with paint, trim and imagination, an artist could contribute ideas to make even the interior corridors of the fictional Enterprise suggest various normal living environments without interfering with the ship's "five year mission". Making an actual space habitat esthetically reassuring would in practice be even easier than dealing with a fictional space ship. A reassuring esthetical ambience would seem vital to the successful long term use of space habitats. One of the most urgent needs is for methods of creating the illusion that a space colony is bigger than it really is—that it is spacious, homelike, and resembles normal Earth living conditions.

Stage illusion is a unique discipline about which most scientists working on the space program are naturally unfamiliar. Nevertheless, the principles of artistic illusions will be vital in making space habitats acceptable to families. While, of course, architects should design the actual buildings, they are unfortunately not trained in the field of illusion, a field of unique importance in regard to a pioneer habitat in space, particularly in view of the inherent distortions.

In addition to detailed suggestions for making the space habitats "homelike", this study has resulted in two unique general conclusions that have not been published elsewhere:

1. No matter how small the space facility is, it can be made to look relatively "normal" and "earthlike" with modest expenditures but careful planning. There is artistic precedent for creating even "outdoor" effects in an enclosed area.

2. If economy or other technical reasons make two or more levels necessary or desirable, it will be virtually an esthetic asset. Surprisingly enough 150' is more than ample height for esthetic purposes for a level in a space colony. In dealing with interior space it is possible to have a ceiling too high.

MAKING THE SPACE COLONY APPEAR BIGGER

Method 1. Cave Dwellers.

One of the techniques a magician uses is to study the audience. The same show is not suitable for church ladies, a stag party and a 4-year old's birthday party. Great care should be taken in selecting material to suit the audience.

To apply this technique to a space colony, note that the colony should appear larger to the *inhabitants*. They constitute the audience. The colony cannot be changed but the colonists can be selected. The conditions on the space colony are not going to appear the same to a man born in Manhattan who has spent his entire life riding the subway twice a day, as to a forest ranger from Minnesota who doesn't see more than 10 people per month in his normal daily activities. The space colony will *be* the same, but it will not *appear* the same to those two individuals. Hire people who have backgrounds that include living in inner cities, a long tour of duty on a submarine, working in coal mines, running an elevator all day long, living in a family of 6 children in a one bathroom apartment, etc. If the inhabitants are accustomed to crowded living conditions, the space colony will appear much bigger and more spacious than it will be to someone who is used to wide open spaces. Hire natural born cave dwellers, not open air lovers. This is most important for the first group of people because they will establish the attitudes of all following colonists.

This technique will be very effective. It won't cost any more or add anything to your budget. It is obviously not possible to hire 100% of the colonists with an inner city background. This is a partial technique. If 60 or 70 percent have this type of background, it will make a significant difference in the happiness of the colonists.

The logic of hiring people accustomed to crowded conditions is clear; the necessity of using techniques that only do part of the job may not be so clear to a scientist. Therefore, it is imperative that it be called to your attention that nearly all esthetic considerations are based on doing part of the work. A hostess places one bouquet of roses in a room; she does not fill every cubic foot of the room with roses.

Method 2. Mirrors.

An old technique for making a room seem larger is adding mirrors on one wall. Mirrors should be particularly valuable on a space colony because they would be low in cost and very high in value. This is an easily available resource that would be of great benefit to the space colony.

Private quarters could have door mirrors, large wall mirrors or even small mirrors to add to the feeling of more space. Public buildings could have large wall mirrors, or mirrored tiles. Restaurants often cover the entire wall above a row of booths to make the room seem double in size. A little strip of mirror in a small garden can add to its effectiveness. A tree placed in a mirrored corner will appear to be four trees.

Not all mirrors are made of glass. Reflecting pools are also a type of mirror. They are usually only a few inches deep and don't contain much water but add a lot to a view of a nice building or a statue or a good garden setting.

Method 3. Color.

Light colors make a room look larger, dark colors make it look smaller. Paint everything white! That is oversimplified, but the more light colors used, the better. Bright colors are more cheerful than dull colors; soft colors are more soothing than harsh colors. One of the strongest emotional weapons is color. It is also one of the cheapest methods of creating a comfortable environment. The esthetic difference between a building with color coordinated walls, fixtures, ceilings, floors, doors, etc. and the inside of the bottom of a battleship painted dull grey all over is painfully obvious. Color coordinating is easy to do when a project is new. It is relatively easy to buy new things that match in color. Renovating an older structure is much more difficult when colors have to be matched. Starting from the beginning, it should be no real trouble to create a pleasant colony with lovely colors.

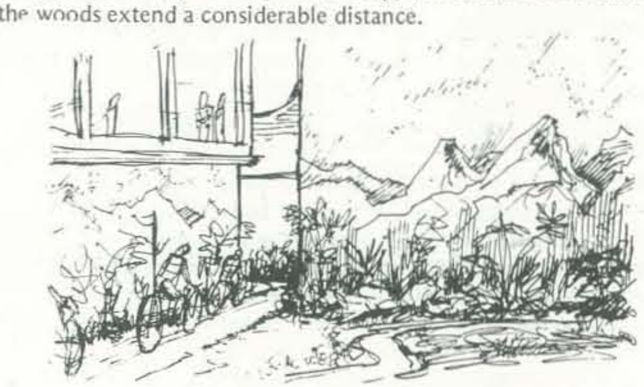
Method 4. Photomurals.

Photomurals are an easy way to add an artificial view to a room especially if you do not have windows. I have a friend who has one on the wall of her kitchen. Her husband added to its effectiveness by putting a real window in front of it with a small light at the top and then curtains at the sides. In stead of staring at a blank wall, she now has a lovely view out a window at an apple orchard in full bloom. This adds more to a room than just hanging up a nice picture. It adds another dimension. The more crowded the colony has to be, the more you need this type of artificial view. It is inexpensive, beautiful and gives the illusion of enormous space.

More expensive and elaborate effects are possible with projected slides. Projected movies of subjects such as skiing in Maine or surfing in Hawaii are delightful and could be added as economics permit, particularly nice in areas of long corridors or dreary waiting areas. This is the type of thing that is flexible. It could be added later, but would be much easier to add if it were planned in advance. Wiring or fittings or space could be added early, and expensive projectors later.

Method 5. Illusion gardens.

There is a restaurant in East Haddam, Connecticut, in the rear of a gift shop. It consists of a row of ice cream tables and chairs in front of a glass wall. The glass wall covers the entire rear wall of the building and is about 1½ stories high. Behind the glass is a beautiful garden with a fountain. At first glance it appears that you are looking out on a lovely wooded area that continues off into the distance. On closer scrutiny it is clear that the garden is only about 9 feet deep. The garden consists of several trees, vines and plants climbing on a cliff and carefully placed plants in the foreground. In a space only 9 feet deep, there is an illusion that the woods extend a considerable distance.



A pirate restaurant, off the shore of Florida, has dozens of aquariums built into the walls. It is decorated in a pirate nautical motif and creates the illusion that you are under the sea looking out port holes at the ocean. These aquariums are only 1 or 2 feet deep.

In New York City, there is an elevated highway running along a row of apartment buildings. One of the apartments has a large balcony. There is a table for two and a formal English garden. Between the balcony and the highway is a fence at the edge of the balcony made of potted evergreens. Although the balcony is only about 6 or 8 feet deep, the trees effectively wall it off from the constant roar of the traffic.

In each of these examples, a narrow strip of space was converted into a garden or scenic view that gave the illusion of a much larger view–a forest, an ocean, a tropical jungle, and a formal English garden. This technique could be used to great advantage either on small balcony areas or at the side of public buildings. It is much more effective than the use of a strip of garden down the center as you have in many shopping malls. It implies that there is much more garden off in the distance.

Method 6. A part implies a whole.

If you see a dog's tail sticking around a corner of a building, you assume there is a dog attached. If a street curves around a corner, one assumes it continues. It may actually end in a small courtyard or deadend, but the fact that it curves out of sight implies that it continues, and makes it seem much longer. Streets should not be long and straight; they should go around corners or curves. This will also block views where the street would normally end too soon. Buildings can also be used to block long views as if one had to go around a corner to see more.

In a very small space facility such as the one described in Vol. 3 in Earth/Space News, consisting of a pod 200' x 300' x 500', or an early construction facility with small dimensions, the buildings should be built against the walls. This will perform three useful functions:

1. Walls will be hidden. Walls remind a person that he is in an enclosed space. Buildings look more Earth normal.

2. There will actually be more room in the center of the facility if the walls can be used as one wall of buildings.

3. Buildings usually have an exit on the other side. Actually these will not have an exit since they are built against a wall, but they will appear to be deeper than they are and to have an exit or even a street on the other side. It will give an illusion of more space than you actually have and will make the small space manufacturing facility seem to be bigger than it is.

MAKING THE SPACE COLONY *APPEAR* MORE OPEN

An enclosed space seems like a prison if people can't open the door and go out. Conversely, if people can go in and out at will, it is not a prison even if it is very small and cramped, and the emotional problem will not be too bad. Much of this is pure emotion. Many people are stay-at-homebodies and don't actually go out very much. But they want to feel that they can go someplace else if they want to.

Method 1. Actually leaving the colony.

First it will be necessary to have contracts that allow people to return to earth if they cannot stand space. This is primarily a legal problem, but esthetically, the door must stand open to return. Sabbatical trips back to earth after a few years would also be excellent. At a later date, it should be possible to have frequent trips to other colonies and to the moon and maybe even other planets. This covers the reality of actually leaving the colony. Other esthetic solutions to the problem are as follows in the next three methods.

Method 2. Contact with home (Earth).

Mail is an obvious contact. Besides letters, there will be heavy mail order shopping. Probably a service for obtaining special items will have to be set up, as shops on a colony will primarily carry standard items.

Phone calls to earth will be very valuable. If possible each colonist might be allowed one free phone call per week to earth. Visual phones will be even better when available. Since most cities have TV equipment, arrangements to send up TV tapes of friends or relatives might be explored.

Visits from other space travelers will be helpful. As the colonies grow and become more successful, traffic will get heavier and there will be a lot of people coming and going, especially going on to the new colonies. This will alleviate the feelings of being cut off and will make the enclosed space seem more open. The first colony may well become a busy port-of-call quite rapidly.

Method 3. Viewing the work and agricultural areas.

In most of the proposals for space manufacturing facilities, work areas and agricultural areas are in separate areas from the living areas. Transportation systems link them and they are often in different gravity zones. This would give you someplace else to go. You will probably need a viewing area anyway for visiting VIPs, Congressmen, the President, etc.; why not make it an excursion area for wives, children, and anybody else who doesn't work in that area? A space would be needed for visitors to view the area, but be kept out of the way of workers and out of the way of danger. Either a glassed-in area or a walkway made of see-through grating would be acceptable, probably hung up above the work to be out of the way and to give a good view. In the agricultural areas, it would be nice to have benches placed so that the greenery could be seen. This would be a good substitute for parks, woods and meadows. Long walkways, even if they were above the farms would be very pleasant.

Workers would probably commute daily but if they worked in the farm area, they might like to visit the factory and vice versa. If the factory area is Zero-G, a recreation area would be worth having when economics permit.

One of the most important principles on a small space colony is making each facility serve more than one purpose. Any space colony will have a work area and an agricultural area. It isn't very hard to set up a method for visitors to view them. It is the only place people can go besides the living quarters for the first few years and it will give a feeling of having been "out" and having gone somewhere else. It will give a feeling of being able to open the door and go in and out. This is a very effective countermeasure against the illusion of being in a "prison".

Method 4. Seeing out.

At some place in the space colony there should be an observation platform where you could see ships dock, see the moon, the earth and the stars. There will, of course, be a scientific study area with telescopes, but the observation platform is for the average person, not the scientists. It is important to be able to see the heavens. It will maintain a sense of perspective; a feeling of reality. A person must know where he is in relation to the universe.

It is part of the problem of being away from the earth, that humans need to be reminded of where they are and where they have come from. To maintain sanity, people must keep in touch with reality.

MAKING THE COLONY LOOK MORE EARTH NORMAL

This is a complex problem. Each type of space colony has advantages and disadvantages. Rather than discuss all of the plans, I will discuss the most economical and spartan.

The most economical suggestion published is a pod of 200' x 300' x 500' designed to hold 5000 people with separate areas for agriculture, services and work. This would be cut into several levels and have artificial light.

Even though the entire colony is man-made, it would be esthetically reassuring to have the esthetic values of night and day, indoors and outside, above ground, underground, sky and ground level, and even some feeling of weather. None of these values are normal in a man-made space colony. All of them can be suggested with illusions, at least enough to lessen cultural shock.

Illusion 1. Indoors and outside.

Streets should be "outside" and apartments should be "inside." One method of creating this feeling is by keeping the temperature of "outside" areas a few degrees cooler than apartments. Proper temperature for plants should be studied. Possibly a further reduction in temperature at night would help.

Heavy emphasis on plants in "outdoor" areas would add to the feeling of going out. Trees, vines, flowers and ground covers could be profusely used.

Even with no weather and no automobiles, outdoor materials such as concrete should be used on streets and materials such as wallpaper should not be used on the outside of buildings.

This brings up some interesting possibilities for the future but at first, appearances should remain normal. The first colony will be difficult and will have enough problems to solve, without being experimental in too much at once.

Illusion 2. Night and day.

Some method of cutting off the light and turning on street lights will create an artificial night. This may not be necessary for humans. Maybe we don't want night, but we should start with a night period and make changes gradually.

Illusion 3. Underground and above ground.

There is no ground in a space facility. Even if there is a farm, the soil could be placed on shelves. It would not be necessary to walk on it. For esthetic reasons, it would be desirable to create a ground in the living quarters. This should actually be a service level 3 meters thick. On top of this level should be placed streets, sidewalks, soil for plants, etc. Inside it should contain all the service pipes such as water, heating for air conditioning, tunnels for servicing pipes (it must be tall enough for workmen to stand up comfortably), sewage and garbage disposal, telephone wires, etc. It would also be deep enough to provide large plots of soil for trees, and would be the right height for basements under the buildings. Basements are useful and practical but also are esthetically pleasing in that they give the illusion of an underground and an aboveground.

Illusion 4. Sky.

In a small pod there would be no sky. In a large cylinder with large glass areas, there would be a good feeling of sky. The ceiling of the pod can be treated to look like a sky. It should be painted white or light blue. A projector with a movie of blue sky and clouds can throw a good illusion of sky on the ceiling. At night, a film of the moon and stars would be useful.

If the pod is cut into several levels, each level should have its own "sky" on the ceiling and a service level to create a ground level. This would be more esthetically reassuring than cutting the pod into partial decks with full open space in the center. Even in a sphere, two or three fully closed levels might be considered, even if they had to be artificially lighted and only the top level had sunlight. Three full levels would increase the amount of working area and useful space by double more more. They could be made attractive and very beautiful if you use an illusion of sky on the ceiling and a service level to create a ground level.

The best artistic examples of the creation of an outdoor area that is completely indoors are Walt Disney's Pirates of the Caribbean ride in both Disneyland and Disneyworld. The one at Disneyland in California starts with a boat ride through a Louisiana Bayou. He uses a night scene with artificial stars, trees, buildings, etc. The ride continues through an underground cave, again completely artificial, and ends with a battle in a cove between a fort and pirate ships and a trip past the captured town. Everything is indoors, but the illusion of being outdoors is perfect.

Illusion 5. Cutting the space colony into levels.

In studying the problem of cutting the colony into levels a surprising fact came to light. It may be an esthetic advantage rather than a disadvantage to cut the pod or sphere into levels. Three problems arise: the need for more useful space; visual distortions caused by curvatures in a sphere, and the fixed visual point of reference caused by interior space.

In a sphere, torus, or cylinder, the curvature causes buildings and people to be seen upside down or at an unnatural slant. How much of a slant is bothersome to humans? Four degrees. That may be surprising but, what is the most famous leaning structure in the world? The leaning tower of Pisa. It leans at a 4° slant. Cutting a sphere or torus into levels would cut off the worst views. If it is impossible to see a long distance, the colony will look much more normal. Humans may be able to get used to seeing an upsidedown world, but it isn't necessary. It would be easier to block the view in a cylinder with an artificial cloud, and it would be a great improvement esthetically.

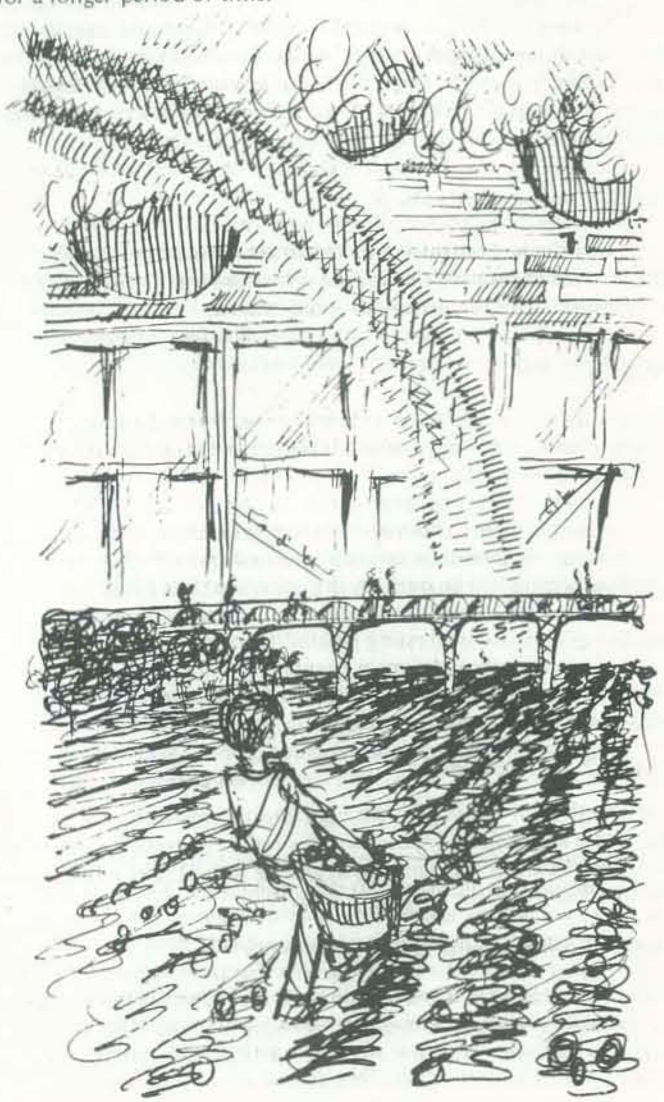
Interior space has a unique problem visually because the ceiling creates a fixed point of reference for the eye. Normally a human being walking around does not look up much. Out-of-doors, the sky has no fixed dimensions unless a bird or airplane flies overhead. The sun, moon and stars are all optical illusions, in size especially. If a bird flies overhead, there is an exact point of reference, and a human will stop, raise his head and look up.
When there is an interior space, there is an exact point of reference; that is, the ceiling, and the same thing happens. A human will stop, raise his head and look up.

Tourists visiting the Vehicle Assembly Building all stop and crane their necks. The VAB is 525 feet high. This height is uncomfortable for people to look at. It is too high to be esthetically pleasing. When they go outside, they do not crane their necks and look up, even though the sky is actually infinitely higher than 525 feet.

Artistic precedents for interior spaces indicate that 7 to 10 feet is normal for the height of a room. Three stories high is the height for a shopping center or a vast library or art museum hall. "Great Architecture of the World" says that "Notre Dame Cathedral in Paris . . . is 110 feet high—the first cathedral built on a truly monumental scale." And "Amiens Cathedral . . . the incredible height of 139 feet." In building St. Paul's Cathedral in London from 1675 to 1710, Christopher Wren found it necessary to build an inner dome and an outer dome because the outer dome would have been too high. It would not have looked right. Grand Central Station in New York Cityis 116' highand 125' long. Penn Station, now torn down, was 150' high. Walt Disney used an interior that is 10 stories high for the inside of his Contemporary Hotel that has a monorail station inside. The most famous artists in the world have used interiors between 100 and 150 feet high to create a feeling of vastness and grandeur in interior spaces. Even in ancient times they used the same dimensions. The Pantheon, Rome, is 143' high and has a diameter of 143'. Anything taller is unnecessary from an esthetic point of view and can even be a disadvantage. Sir Christopher Wren went to a lot of expense and trouble to add a lower dome to St. Paul's in London. The VAB with its height of 525 feet is not noted for its beauty. If larger interior spaces are needed for engineering purposes, that is necessary; but if larger spaces are being left for esthetic reasons, that is not needed.

Illusion 6. Apartments in small spaces.

What are the smallest spaces that humans live in on a voluntary basis? College dormitory rooms, army barracks, small apartments, cabins in cruise ships, houseboats and trailers. The only really small one of these that people live in for long periods of time is a trailer. Many people live in trailers for years; therefore, the technology that has been developed for trailer living should be examined for application to tight living in a space colony. In a space 7' high, 8' wide by 14' long a trailer may include a separate bathroom with shower, medicine cabinet and toilet; refrigerator with freezer compartment, closet, stove, oven, sink, dining table, overhead storage cabinets, 5 windows, 1 door, bunks for 6 people, and a space heater. If a trailer can hold so much, presumably one or two people could be comfortable in a similar type apartment for a longer period of time.



The best thing about trailers is the efficient use of space. All furniture is built-in and much of it folds or slides out of the way. The use of this technology would mean inhabitants would not bring their own furniture. This would result in great savings in space, weight, and cost, not to mention the advantage of interchangeable parts in case of breakage.

Of course, if people live in a tiny space , they will want to go outside. Balconies are virtually a must; not a 2' balcony, but an 8' x 10' balcony.

There are esthetic values in a balcony over a room. A balcony is outside space and can contain a garden. Although it is private outside space, it is still part of the outside area and does not subtract from the open space as a room would. One published plan called for 50% open space and 50% enclosed. A large part of the 50% open space can be cut into private balconies and still maintain the illusion of open space while giving people a "front yard" including a garden and a feeling that they can go in and out of their apartment.

With tight living conditions, soundproofing should be given a great deal of attention. Privacy is important.

MAKING THE SPACE COLONY *APPEAR* MORE HOMELIKE

Method 1. Family.

What is the one thing that means home to most people? Family. The single most important esthetic factor in keeping the workers permanent rather than transient is having a relative or good friend on the colony. In all permanent migrations in human history (other than forced such as penal colonies on Australia or slaves), families have migrated together. This means the extended family, not just the nuclear family. If the space colonies are to become permanent colonies, this process must receive some priority. Human and esthetic values are important in making people happy long range. They *need* their extended family.

Method 2. Vistas.

"Vista suggests a view seen through a long narrow passage as between rows of trees." Webster's Dictionary. The difference between a view and a vista is important esthetically. All long views will look distorted on space colonies but vistas can be lovely. For instance, many people have had the experience of walking down a busy city street and suddenly coming on a small courtyard framed by a wrought iron gate with a little fountain or wishing well. The charming narrow streets of Beacon Hill, Boston, Massachusetts, are famous.

Another type of vista is the favorite of landscape gardeners who like to arrange a statue of a girl with a bird next to a birdbath and a few shrubs to fill a corner of a garden. Oriental gardens are noted for their ability to create exquisite vistas in a small space.

Method 3. Illusion rooms.

In the Musee Grevin in Paris there is an illusion theatre where the audience sits down in an ancient Chinese temple. Presto and the walls and ceiling rotate and you are in a jungle. Presto and they rotate again and you are in a Hindu temple from India. This is a more complicated effect than should be recommended for a space colony, but it is used to illustrate the point that it is possible to create the illusion of being in several parts of the world in one room.

In 1964, the Coca Cola Company built an exhibit that allowed you to walk around the world in about an hour. It was a walk-through exhibit actually all in one building, not more than 40 feet high. First, visitors walked into a street in Hong Kong, very realistic with tiny shops and crooked streets, then they went into a Cambodian jungle past the ruins of Ankor Wat. Next came India, past the Taj Mahal. This was followed by a ski lodge in the Swiss Alps with a view of the alps out the window. It was cold. Then visitors walked onto a deck of a cruise ship, (the vibrations of the engines made the deck appear to be in motion) in the harbor of Rio de Janerio with delightful views of the mountains and the city.

Each of these very realistic exhibits took the space of one large room. They could be reproduced as pocket parks.

Two of the greatest botanical gardens in the world are the Missouri Botanical Garden (Shaw's Garden) in St. Louis, Missouri and the Palace of Plants at Meise, Belgium. At Shaw's Garden, there is a Climatron that reproduces microclimates of every climate in the world. The Climatron is basically a geodesic dome. The Palace of Plants has 2½ acres of greenhouse, the largest in the world, containing 35 rooms of plants. Both of these could be contained in a large space colony and could be used as pleasant places to see beautiful plants as well as scientific study areas.

Method 4. Dioramas.

At a later date, dioramas such as the ones in the Natural History Museum in New York would be beautiful and would add to the feeling of spaciousness. These could fit into the corners of corridors or public buildings. They are expensive but do not require large amounts of materials. Instead, they require the workmanship of a fine artist.

Method 5. Pocket gardens and parks.

Large parks and wooded areas might not be practical on the first space facilities. Pocket size parks and gardens will be relatively easy to plan and implement over a period of time. In New York City there are small parks tucked away on busy streets that are delightful. One has a waterfall built on the side of a building (recirculating, of course), several small trees, and little tables and chairs. Small parks with horseshoes, croquet, bocci, checkers and shuffleboard and darts would be easy. Formal fountains, wishing wells, grape arbors, and beds of begonias, do not take much space and form a pleasant place to sit or walk.

Method 6. Themed Parks.

On a larger space colony with more room, a lesson might be learned from Disneyland. For variety, even a small colony might have various ethnic or imaginative areas. It is possible to have streets with different decor. Oriental, Mexican, German and Italian to give a few examples. These streets could have restaurants, theatres, and shops with appropriate specialties and decorations. On a still tighter budget, each building could be different as is done at world fairs. In a very small colony, variety is more important than continuity. Washington, D.C. and Brasilia are built with a grand plan and are beautiful; but in a tight situation, Greenwich Village, New York; SoHo, London; Montmartre, Paris; or Vieux Carre (French Quarter) New Orleans are more appropriate.

Method 7. Use of Lunar materials for decorative purposes.

GLASS. Fiberglass draperies should be made in bright colors even if dye has to be imported from earth.

In Zero-G glass may be used in ways not possible in normal gravity such as giant bubbles or other strange effects.

Colored glass and stained glass will add cheerfulness.

Prisms hung in front of a light will make the area more colorful and less dreary. Cut glass and beveled glass will also produce rainbows but are harder to make than prisms. Even such tight quarters as Skylab could have a prism.

TILES. Both ceramics and glass for glazes are available. Tiles would be excellent finish materials for homes and buildings. Pottery and decorative ceramics would be good hobbies.

BY-PRODUCTS. If rabbits and chickens are raised for food, there would be feathers for pillows and decorations and rabbit fur for rugs, etc.

METALS. Wrought iron balconies are common in New Orleans. Many decorative effects are possible using patterns already available.

MISCELLANEOUS SUGGESTIONS

1. Winding narrow pedestrian streets without automobile traffic lend themselves to cozy cities like the old Medieval ones with buildings close together and even second story overhangs.

2. Office workers today see very little sunshine—new all air-conditioned buildings are often blank walls on the outside so life without real sunshine won't be too strange if it is necessary.

3. Set-backs. If you build a small space station such as a Pod, or other system with levels, the top story should not appear to reach the roof. It can actually reach the roof and probably should for structural reasons but it is relatively simple to appear to stop short. Either a set-back or a decorative cornice will cut off the view and make it appear to stop short of the top.

4. Since there will be no weather, all plants will have to be watered by some type of sprinkler. If sprinklers are located in the ceilings, this could look like rain. It could be set to fall only on certain areas or be timed for an early mornings hour like 4 a.m.

5. Plants. Even decorative plants could serve two purposes. They could be either useful or edible. Strawberries and violets are good substitutes for grass and are edible. Bamboo shoots are edible and it is very useful as a wood substitute. Honeysuckle vines make good baskets. Grapes are edible and the vines are decorative. Trees use very large amounts of water; they might be limited to fruit or nut trees, plus one evergreen in a central position for a Christmas tree. Plants can be grown in windowboxes, on poles and hanging pots.■

REVIEWS

DIGITAL TROUBLESHOOTING
by Richard E. Gasperini
Hayden, 1975, 180 pp., $9.95
Available through People's Computer Company Bookstore

Many computer hobbyists are either neophytes or "software people" who may have had little or no experience in the care and feeding of LSI hardware. When their JOLT, KIM, or POLY breaks a leg they either ship it back to the factory, tote it to the nearest computer store, or shoot it. There is, however, a book that should be quite helpful in providing a basic understanding of digital troubleshooting techniques. This is Richard E. Gasperini's *Digital Troubleshooting: Practical Digital Theory and Trouble-shooting Tips.*

This book introduces the reader to the fundamentals of bipolar and MOS technology, explains how to read logic diagrams, and discusses basic tools and diagnostic methods. It then covers the use of common digital circuits such as decoders, multiplexers, flip-flops, buffers, counters, and shift registers. The book does assume some prior knowledge of electronics, but not of digital circuitry. It discusses the use of memory and display devices, and gives many useful tips on soldering and unsoldering ICs, obtaining replacement parts, etc. Chapter 18 contains some good suggestions on obtaining additional information, and an appendix explains how to interpret manufacturer's part numbers, such as N74LS195N.

Reviewed by Jim Day.

PROBLEMS FOR COMPUTER SOLUTION
by Steve Rogowski
Educomp Corporation, 1975, 271 pp., $9.95

When asked to review the book *Problems for Computer Solution,* I thought it was just another elementary programming book. But when I started to get into some of the problems for which you can write programs, the book provided an interesting challenge. Instead of *giving* you everything to work with, *you* must look up the formulas and ideas around which to write the programs to solve the problems. There are 90 problems which go from simple number conversions to Einstein's energy equation. This book will challenge even the best of programmers.

Reviewed by Glen Jensen.

MICROPROCESSORS: TECHNOLOGY, ARCHITECTURE, AND APPLICATIONS
by Daniel R. McGlynn.
Wiley, 1976, 207 pp, $11.95.

This book tries to present a comprehensive, well-thought-out, well-organized overview of the economics, applications, manufacture, architecture, interfacing, and programming of microprocessors in only 187 pages! Sad to say, it's impossible. I really like the selection of topics and the scope of the book, but the technical rationale is so terse. I'm afraid that the expert will learn nothing and the beginner will conclude that he just can't get it. My favorite chapters were the first two – essays giving an overview of the field and dealing with broad topics such as system design and market impact.

Reviewed by Larry Press.

MATHEMATICAL CARNIVAL
by Martin Gardner
Alfred Knopf, 1975, 274 pp., $8.95
Available through People's Computer Company Bookstore

Martin Gardner is well known among persons interested in recreational mathematics for his excellent "Scientific American" column. Previously he has come out with six book collections from his column. Now there is a seventh book – *Mathematical Carnival.*

Whether your interests tend to puzzles, tricks, games, or art, there is something here for you. In this marvelous volume you will find sprouts, transfinites, hypercubes, calculating tricks, M. C. Escher prints, card tricks, the origins of biorhythms, superellipses, Pascal's triangle, how you cook a puzzle, square packings, and numerous puzzles.

This book is a valuable addition to the library of any puzzler or recreational mathematician.

Reviewed by Eryk Vershen.

BOOK BUYERS

People's Computer Company has a bookstore: our aim is to make books about computers readily available to you. If you subscribe to *People's Computers,* you'll soon be receiving the latest update of the *People's Computer Company Categorical Bookstore Catalog.* If you'd like to receive extra copies, drop us a line. And let us know about books you think that the bookstore should carry but doesn't. Thanks –

Dan Rosset
People's Computer Company
Box E
Menlo Park, CA 94025

ONCE UPON A FAIRE

Were you at the Faire?
The Festival? The Fest?
What forecasts do *you*
make for hobby/home/
personal computing?

BY TIM BARRY

PHOTOS BY INTER GRAPHIC

In the beginning there was sand. And lo, the great manufacturers of silicon gulch did take the sand and labor mightily to bring forth the transistor. And the transistor was fruitful and multiplied into RTL, DTL, TTL, ECL, and all manner of MOS devices. And the crowning glory of them all was the microprocessor, whose powers did bring stars of delight to the eyes of industrial manufacturers everywhere. And while the industrialists did design the microprocessor into every tool and equipment imaginable, a group of hobbyists from the land of homebrew did rise up and say, 'Surely we, too, can do mighty things with the wondrous microprocessor.' And the industry did laugh and scorn them, for surely hobby computers could never be of any real interest. But the hobbyists were not discouraged and they strove to forge a niche of their own. Soon all manner of strange and interesting hardware, software and magazines did begin to appear, most produced by companies with strange names. After much struggling and wandering in the wilderness, it came to pass that Jim Warren, the high priest of Dr. Dobbs, did decree, "Let us meet in the city by the bay in the month of April, on the day of all taxes to celebrate and show the world our new found market." And the word went out and the hobbyists came from far and near to celebrate the great faire. And the temples of industry were shaken, for they knew now that the time of change was upon them. And it was good.

All allegory aside, the first Annual West Coast Computer Faire had to be a milestone of some sort. It drew over 12,000 people (official total attendance = 12,755) and over 160 exhibitors from all over the country. It had a series of extremely interesting technical sessions and, as something of a first for computing exhibits, the Faire had an entire floor devoted to exhibits of personal computer projects, and most of these were very impressive. In terms of size and scope the Faire was every bit the equal of the large national computer shows.

I could try to do a 'who, what, where' summary of the Faire, but I will resist the temptation. First of all, you really had to be there to experience the crush of the crowds, the bodies stacked three deep at the exhibitor's booths, and the overall feeling of shock that so many people were there. Anyone who questioned the interest in personal computing surely had his doubts crushed along with his toes.

Second, there was simply so much there that any responsible coverage would take lots of pages. I shall, therefore, opt for the totally arbitrary path of some captioned pictures of those things which caught my eye. To those exhibitors not mentioned (and there are plenty), my apologies.

Finally, there were some interesting undercurrents at this show which I think deserve a bit of commentary. If my feelings are correct, we may be able to extrapolate some of these undercurrents into trends that the hobby market is going to follow: I should like to make this analysis in three areas:

1. The systems exhibited
2. The cross section of the exhibitors
3. The nature of the crowd.

Mostly these are 'vibes', if you will, and I would certainly like to hear from those of you who were tuned to a different channel.

In terms of hardware presented, the show offered us a side by side look at most of the 'new' systems. We have now had about two real years of commerical computer hobby manufacturers. In that period of time we saw a handful of manufacturers emerge as the front runners in the business. Most started small, so to avoid high development costs a great many latched onto the S-100 nee Altair bus with great haste and charged. Almost overnight these companies produced a proliferation of somewhat compatible hardware building blocks to fill almost any need.

Well, now the smoke has cleared and we find the hobby business pretty well in the hands of these few manufacturers. Sure, lots of small guys make a special card or two, but the mainframes are pretty much coming from the same places. All these intially successful firms have had to face the same, 'What do we do for an encore?' question. It is no longer enough to be simply duplicating the same basic CPU, RAM, ROM, and I/O cards. Instead, the companies must now have some integrated system visions and, to be successful, this vision must match the market place. From a hobbyist standpoint, I think these second generation attempts can be aptly described as winners and losers.

The systems most likely to succeed (i.e. sell) in the hobby market are made by those manufacturers who are attempting to draw their hardware and software together into modest performing, reasonably prices, well-documented, well integrated systems. 12-16K of memory, T.V. interface, keyboard, cassette and maybe some floppies will be the hobby baseline for some time. The mass hobby market is probably not interested in buying mega byte memories, massive chassis with three cubic feet of air, and the generally expensive ($5000 up) turnkey systems. Manufacturers who are headed this way are headed out of the hobby world and into the small business systems world. This places them in a direct line for a confrontation with the minicomputer business, and good luck to them when that happens. In that type of marketplace, the peripherals, software, and support make or break the business, and so far the micro track records in these areas have been less than spectacular. Small business computing will undoubtably get here, but it will become increasingly divorced from its hobby oriented origins.

The cross section of exhibitors present provides a very interesting look into where the personal computing field is right now. I wandered around the main floor and tabulated the majority of exhibitors as follow:

Computers, Computer cards	36	Instruments	1
Computer Peripherals		Publishers	
Floppy Disk	5	Book	2
Printers	3	Magazine	13
Terminals	9	Computer Stores	12
Cassette Readers, Punches	4	Software Vendors	4
Music and Speech	4	Computer Education	4
Components	8	Miscellaneous Exhibits	13
Tools	2		

While there is obviously some overlap among these categories, I think they are still interesting. Most obvious is that over 2/3 of the classified exhibitors are selling hardware or hardware related products.

Of the hardware exhibited, 8080's and Z-80's in S-100 bus configurations seemed dominant, with S-100 bus peripheral cards being overwhelmingly the most common. The available peripherals were pretty much well known, although, there were three fairly new low cost (under $1000) medium speed printers.

There were only four (less than 4%) independent software vendors selling released products independent of the hardware systems. There were also only four vendors attempting to teach people how to use the systems available. Of these, two were directly tied into fairly specialized training hardware which has only limited usefulness after the training ends. When you consider how long and loud people have been making noises about how important software and education are to the overall success of personal computing, it sort of makes you wonder. In fact, this fact leads me to the last, and probably most important observation of the show: people aren't really buying yet.

I saw crowds of people picking up and buying literature by the pound, but I suspect very few people walked away with a new computer or even a real commitment to buy. The large popularity and sales of the magazines and books show people are really interested in learning about the equipment, but they're not ready to plunk down the dough. Most people now seemed aware that this hobby takes more technical skill than your garden variety consumer hobby, and they also seemed to wonder whether or not the high initial cost would ultimately be justified.

The interest in personal computing is real. Of that there can be little room for doubt. But until the quality of education and applications software reaches the point where the average technical minded person perceives he has a reasonable chance of constructing a useful system; the interest will probably not be converted into sales. I think the real challenge is not who can do the newest, fastest, neatest hardware. Rather, it is who can best construct a system which teaches people to use the already considerable power currently available. This will be an expensive job, but if personal computing is to emerge as a major hobby rather than a technical curiosity, it will have to be done, and the sooner the better.

1)There were a good many talks, exhibits, and products based on computer speech synthesis, voice recognition, and music generation. Shown here is the Logistics Speech music synthesizer, running in an Altair 8800b microcomputer.

2)Dr. Franz Frederick both talked about and demonstrated his implementation of "Waldo, the computer controlled turtle". Waldo's design and control language will be the subject of a series of articles Dr. Frederick plans to write to introduce robotics to the hobbyist community.

3)Scientific Research was one of the few vendors at the faire who were selling software independent of hardware. Their BASIC Software Library seemed to be drawing the interest of many users who have specific plans for their systems.

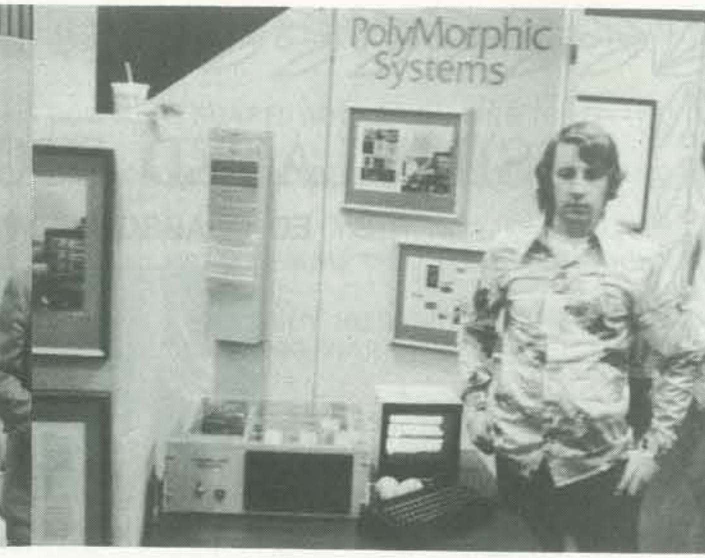
1) Ever vigilant, the dragon lined up new computerists for PPC.

2) The University of San Francisco had one of the larger home brew exhibits. Shown here is their PDP-11 based computer music generator.

3) The crowds at the faire were large at almost all times. Here is a view of the main floor less than an hour before the show closed on Sunday.

4) Onlookers crowd on up to get a look at the latest Apple computer hardware.

5) Digital Systems and Digital Research shared a booth at the faire. Digital Systems manufactures a highly regarded full ized floppy disk system, while Digital Research developed CP/M, one of the industry's best known disk operating systems. Dr. Gary Kildall, designer of CP/M is shown here answering questions for some interested hobbyists.



8) Many excellent technical discussions and sessions covered topics from computer art and music to legal implications of personal computers. Most sessions pulled a good sized crowd, like this on multi-tasking environments for home computers.

6) Logical Services was one of the few exhibitors selling training material independently of specific hardware trainers. Their Modu-Learn^{tm} *software course seemed to be attracting the interest of many.*

7) PolyMorphic Systems displayed much of their new hardware, including this triple Mini-floppy system. There is little doubt that floppy-based mass storage systems will become more pppular.

9) Tom Pittman, designer of one of the most famous of all Tiny BASICs, was on hand with his Itty Bitty Computers company. He also gave a talk with Bob Davis on the design of a variable architecture computer.

10) The Digital Group's hardware display included this new, low-cost printer. Low-cost hard copy is one of the major gaps in the hobby hardware market: we'll probably see more of this type of peripheral.

A SIMULATED STOCK EXCHANGE

BY ED PEARSON & MIKLOS VASARHELYI

Reading about the stock market is a pretty dry way to figure out what it's all about. And investing your own money to learn how the market works may be an expensive educational experience. Here's an alternative way to explore the market.

Students at the University of Southern California School of Business can learn about the practices and operations of the stock market, and can try their luck (on a limited scale) in stock trading, by utilizing the computerized SIMSEX (Simulated Stock Exchange) program developed by Professors Miklos Vasarhelyi and Ted Mock and Mr. Sam Allen of the Department of Accounting at USC.

The SIMSEX system was designed as a research tool to attempt to understand the decision-making process, especially in regard to stock market analysis and performance; students act as the "traders" in the market whose decisions are monitored so that the researcher can discover the type and amount of information used, and the method in which it is used in the decision process.

From the student's viewpoint, the simulation is a valuable learning experience. The student can trade freely in whichever stocks he or she finds attractive. In doing so, the student is exposed to the terminology of the market and to the practices and operation of major financial institutions and corporations. The student receives ample "money" to invest (at least $100,000 in SIMSEX dollars) and then buys and sells the fictional stocks of real companies at the actual prices of the real stock exchange.

The student further is given the option of receiving more SIMSEX-trading dollars by investing real money into the game (up to $20). Rewards are then given based on the performance which the investing student achieves. For instance, if a student invests $20 and receives for trading purposes an opening cash level of $200,000 and by the end of the game has built a "portfolio" worth $250,000, then that student would receive $25, the equivalent amount of "real" dollars.

Many aspects and features of the market are reproduced in the simulation. Players can buy, sell, sell short, trade on margin, invest their money more "safely" in savings accounts, earn dividends, receive stock dividends and stock splits. They must also pay commissions for their broker's services. Because every stock listed on any major exchange is eligible, the player has a wide range of choices and can review the past performance of companies and make forecasts of their future potential, thus refining the skills they have acquired in their business education.

The earliest versions of the simulation were programmed in the APL language. The expanded and revised version is programmed in BASIC (in the HP2000F and 2000 Access versions); it is a conversational on-line system in which the user speaks with his "broker," Hal Pearson (the computer) by typing at a terminal. By answering Hal's questions, the user completes his stock transaction and, if he chooses, receives a report of his portfolio status, including the amount of cash. The user must know certain information in order to trade; for example, in the "buy" transaction, the user must know the stock name, the stock "code" used by the exchange (as found in any stock guide, such as Standard and Poor's), the stock exchange it is traded on, and the selling price (of the previous day, as quoted in the newspaper's financial section). Before any transaction is finalized, an "auditor" must check the accuracy of the information provided by the player. If the auditor approves, the trade is recorded and the player's portfolio updated. The auditor usually performs his function daily, reviewing all transactions which occurred after the previous audit.

The simulation is usually included as part of the required work in certain courses (although students still have the option to invest their own money or not), so the trading period is a rather short one, one semester. Even so, the user is given an excellent opportunity to observe the stock market and to become familiar with stock market methods and with many of the companies whose stocks are traded.

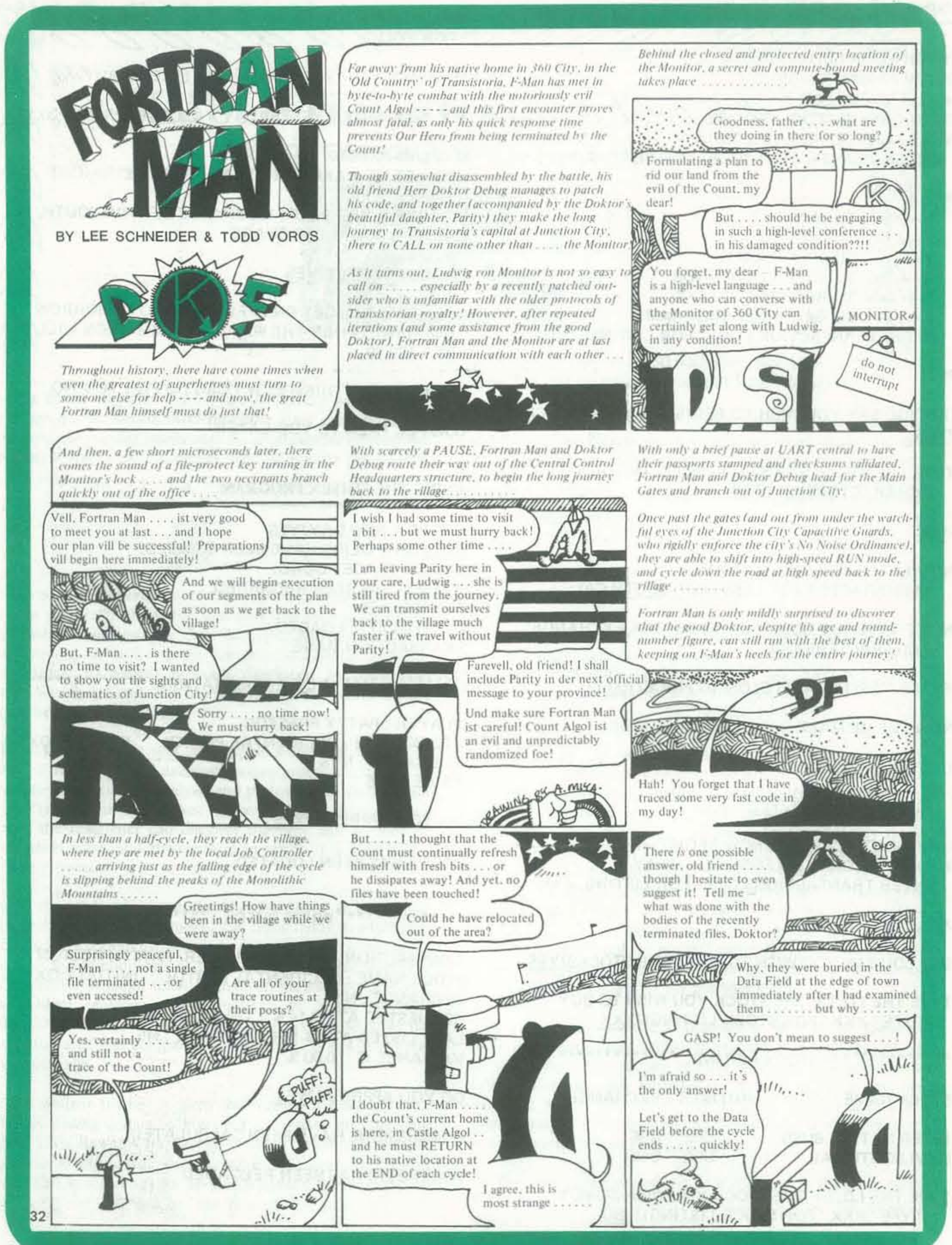
The following runs of the SIMSEX and AUDIT programs indicate the type of student interaction that occurs. Student input in each program is underlined.

USING THE AUDIT PROGRAM

```
ENTER YOUR I.D. NUMBER?1
ENTER PASSWORD?GREGG
HELLO SAM ALLEN
I'M HAL YOUR STOCKBROKER
WHAT CAN I DO FOR YOU TODAY?
  (ENTER TRANSACTION CODE-FOR LISTING, TYPE '0')

?0
   1 ------ TO BUY
   2 ------ TO SELL
   3 ------ TO SELL SHORT
   4 ------ TO ENTER DIVIDENDS
   5 ------ TO REGISTER STOCKS
   6 ------ TO EXIT FROM PROGRAM
  (ENTER TRANSACTION CODE-FOR LISTING, TYPE '0')

?5

DID YOU SAY YOU WISH TO REGISTER A NEW STOCK?
?YES

TYPE THE STOCK'S TICKER-TAPE-I.D. INITIALS
   5 CHARACTERS OR LESS

?USC

NOW ENTER THE STOCK'S NAME
   18 CHARACTERS OR LESS?UNVI.SOUTH.CAL

NEXT TYPE THE CODE FOR THE STOCK EXCHANGE
   3 CHARACTERS?OCE

ENTER YESTERDAY'S CLOSING PRICE?100.13

NOW LET ME MAKE SURE I GOT IT STRAIGHT

UNVI.SOUTH.CAL        USC        OCE        100.13

   IS THIS CORRECT?YES

UNVI.SOUTH.CAL     IS NOW REGISTERED
WHAT ELSE CAN I DO FOR YOU TODAY
  (ENTER TRANSACTION CODE-FOR LISTING, TYPE '0')

?1

DID YOU SAY YOU WISH TO BUY SOME STOCKS?YES

TYPE THE I.D. OF THE STOCK YOU WISH TO BUY
  (TYPE ' XXX ' FOR STOCK LISTING)?XXX

************* STOCK LISTING *************

STOCK NAME                 INITIALS   EXCHANGE

INTER NATION BUSD             IBM     NSE
UNVI.SOUTH.CAL                USC     OCE

TYPE THE I.D. OF THE STOCK YOU WISH TO BUY
  (TYPE ' XXX ' FOR STOCK LISTING)?USC

YES, I CAN BUY SOME SHARES OF UNVI.SOUTH.CAL
  FOR YOU

HOW MANY SHARES WOULD YOU LIKE TO BUY?100

O.K. NOW ENTER YESTERDAY'S CLOSING PRICE?100.13

O.K. SAM ALLEN
NOW LET ME MAKE SURE THAT I GOT IT STRAIGHT

YOU WANT ME TO BUY 100 SHARES OF UNVI.SOUTH.
CAL AT $ 100.13 PER SHARE

IS THIS CORRECT?YES

GREAT!!!  I'LL GET ON IT FIRST THING TOMORROW
  MORNING. OH, BY THE WAY, MY COMMISSION WILL
  BE $ 139.39

IS THERE ANY THING ELSE I CAN DO FOR YOU?NO

O.K. I'LL TALK TO YOU LATER
```

USING THE SIMSEX PROGRAM

```
IS THIS A NEW DAY?YES
ALL PORTFOLIOS HAVE BEEN SUMMED.
END OF TRACE FOUND.
'PRICES' HAS BEEN RECORDED.
PLAYER #22
PORTFOLIO IS LOADED.
TRANSACTION MADE.

************* NEW REGISTRATION *************

PLAYER : PATTY PRESTON
STOCK NAME : OCCIDENTAL PERTOL      INITIAL : OXY
EXCHANGE : NYS
PRICE : 14.5

DO YOU APPROVE?YES

STOCK HAS BEEN LOCATED

************* PRICE CHECK *************

TRANSACTION - 1             PLAYER: PATTY PRESTON
STOCK NAME : OCCIDENTAL PERTOL     INITIALS:OXY
EXCHANGE : NYS
REQUESTED AT : $ 14.5    ON DATE : 345
LAST LISTED AT : $ 14.5  ON DATE :345
VARIANCE IS :  0.00 %

DO YOU APPROVE?YES

COMMISSIONS HAVE BEEN CALCULATED

PORTFOLIO HAS BEEN RELOADED
```



SUBSCRIPTION INFORMATION

Here are a few highlights from back issues that are still available:

Vol 1, No 1: Buying Guide for Educational Computers; Strings; Huntington Project Simulations; games programs; Number, Letter

Vol 1, No 2: 'What's Wrong with BASIC?'; BASIC Music; Terminal Technology; games programs: Stars, Bagels

Vol 1, No 3: Shopping for a Computer; Stacks; BASIC Music; games programs: Chomp, Trap

Vol 2, No 5: More Hardware Descriptions; Project Solo; games programs: Button, Abase, Mandala

Vol 3, No 1: FRED: a Low-Cost Home/School Microprocessor System; Cybernetic Theatre; games programs: Snark, Hurkle, Mugwump

Vol 3, No 4: Chip Talk; Build Your Own BASIC; Random Numbers; Biorhythm Program

Vol 4, No 3: Tiny BASIC; Hardware Guide; the game Number in 3 kinds of BASIC and 8080 assembly language

Vol 4, No 4: Soloworks Curriculum; Computer Games by Mail; space games programs: Sttr 1, Motie, Rescue

Vol 4, No 5: A 2-Terminal STar Trek Game; Huntington Project Simulations; games programs: Dodgem, Square

Vol 4, No 6: Stacks and Logical Expressions; Home TV as a Terminal; Kids Building Computers; games programs: Factors, Pounce

Vol 5, No 1: The Computer Awareness Van; Electronic Projects for Musicians; Calculator Resources; Tiny BASIC; games programs: Tiny Trek, Sinners

Vol 5, No 2: STAR TREK Documentation; Classroom Computer Games; Classroom Computer Building; 'Forget Me Not' by Isaac Asimov; game listing: Planets

Vol 5, No 3: Computer Technology in Schools; PILOT 73; Tiny BASIC; games programs: Hats, Frogs

Vol 5, No 4: Tiny BASIC; Tiny PILOT; 6502 Assembly Programming; Pet Robots; game program: Mastermind

Vol 5, No 5: Z-80 PILOT; Tiny BASIC; 6502 Assembly Programming; Math Drills; game program: Zot

omputers
ion Form

ription to *People's Computers*
) for $8.

☐ This is a renewal
(Please attach your label)

____

____

__________ ZIP________

ble on page 2) H

• Menlo Park, CA • 94025

omputers
tion Form

ription to *People's Computers*
y) for $8.

☐ This is a renewal
(Please attach your label)

____

____

__________ ZIP________

ble on page 2) H

• Menlo Park, CA • 94025

ue Order

k issues of the newspaper
outside flap to the left.)

issue: Vol. ______
Vol. ______
s Vol. ______
Vol. ______
Vol. ______

SUBSCRIBE NOW!

if you already subscribe, please pass these on to friends...

and tell them about our new name and *magazine* format.

Coming Soon:

*how to write computer-assisted instruction (CAI); with examples in PILOT;

*new computer applications: home climate control, biofeedback monitoring services and more;

*how to program in easy-to-use computer languages (for beginners);

*more on the household pet robot;

*programming tips for the HP-25 Calculator;

*student records subroutine for CAI lessons in Extended BASIC;

*programs, book reviews and games.

pc is published by People's Computer Company, a non-profit, educational institution.

people's computers
1263 El Camino Real
Box E
Menlo Park, CA 94025

people's computers Subscription Form

Please send me a one-year subscription to *People's Computers* magazine (published bi-monthly) for $8.

☐ Payment enclosed ☐ This is a renewal (Please attach your label)

☐ Bill me

NAME ______________________________

ADDRESS ______________________________

CITY/STATE ____________________ ZIP ________

(Foreign rates available on page 2) H

1263 El Camino Real • Box E • Menlo Park, CA • 94025

- -

people's computers Subscription Form

Please send me a one-year subscription to *People's Computers* magazine (published bi-monthly) for $8.

☐ Payment enclosed ☐ This is a renewal (Please attach your label)

☐ Bill me

NAME ______________________________

ADDRESS ______________________________

CITY/STATE ____________________ ZIP ________

(Foreign rates available on page 2) H

1263 El Camino Real • Box E • Menlo Park, CA • 94025

- -

Back Issue Order

☐ Send me the following back issues of the newspaper (contents are listed on the outside flap to the left.)

I am enclosing $1 for each issue: Vol. ______ Vol. ______ Vol. ______ Vol. ______ Vol. ______

☐ Send me ALL 15 back issues for only $12.

NAME ______________________________

ADDRESS ______________________________

CITY/STATE/ZIP ______________________________

Mail with payment: ______________________________

FIRST CLASS
PERMIT NO. 756
MENLO PARK, CA

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

people's computers
1263 El Camino Real
Box E
Menlo Park, CA 94025

FIRST CLASS
PERMIT NO. 756
MENLO PARK, CA

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

people's computers
1263 El Camino Real
Box E
Menlo Park, CA 94025

Mail to: people's computers back issues
1263 El Camino Real
Box E
Menlo Park, CA 94025

Please enclose payment with your order.

MORE TINY BASIC

by The Dragon

Copyright © 1977 by Bob Albrecht

Last time, you may recall, we ended with two programs for *you* to write. Here are *our* solutions – yours may be different, yet still be correct.

• ADDITION PROGRAM

```
100  PRINT "IF YOU WILL ENTER VALUES OF A AND B,"
110  PRINT "I WILL PRINT THE VALUE OF A+B."
120  PRINT
130  INPUT "A=";A
140  INPUT "B=";B
150  PRINT "A+B="; A+B
160  GO TO 120
```

• ARITHMETIC PROGRAM (+, –, *, /)

```
100  PRINT "IF YOU WILL ENTER VALUES OF A AND B,"
110  PRINT "I WILL PRINT THE VALUES OF A+B, A-B, A*B, AND A/B"
120  PRINT
130  INPUT "A=";A
140  INPUT "B=";B
150  PRINT "A+B="; A+B
160  PRINT "A-B="; A-B
170  PRINT "A*B="; A*B
180  PRINT "A/B="; A/B
190  GO TO 120
```

Remember, the statement: 150 PRINT "A+B="; A+B tells the computer:

(1) Print the string A+B= on the screen.
(2) Compute the value of A+B and print it on the screen.

THE CAPRICIOUS PROFESSOR RND

Computers are fun, especially when programmed to play games. Many games depend on the computer's ability to generate "random numbers," numbers which appear to be chosen "at random." So . . . just to make things easy, Tiny BASIC provides a special function to compute "random numbers." These numbers are not really chosen at random, but they are unpredictable enough to be interesting and entertaining. Here is a short program to print the numbers 0 and 1 at random.

```
10 LET R=RND (0, 1)
20 PRINT R
30 GO TO 10
RUN
```

RND (0, 1) will be either 0 or 1, at random

```
1
0
0
1
1
1
1
0
1
0
0
1
0
•
```
cursor

You can get a random sequence of ones and zeros by flipping a coin. If the coin comes up HEADS, write down a 1; if the coin comes up TAILS, write down a 0.

IMPORTANT NOTICE: Unfortunately, most Tiny BASICs have a slightly different RND function. Remember, in this series, we are presenting Tiny BASIC as we would like to see it designed.

We interrupted the computer here. The computer prints a cursor to indicate it awaits your next desire.

If we RUN the program again, we will probably get a different sequence of ones and zeros.

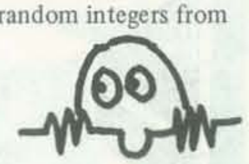
The *function* RND (0,1) tells the computer to compute a random integer in the range 0 to1, inclusive.

The statement 10 LET R = RND (0,1) tells the computer to compute a random integer, 0 to 1, and then assign the random integer as the value of R.

Here is a program to generate and print random integers from 1 to 6, inclusive.

```
10 LET R=RND (1, 6)
20 PRINT R
30 GO TO 10
```

RND (1, 6) is a random integer from 1 to 6.



To generate random digits, use RND (0,9) and to generate random integers from 1 to 100, use RND (1,100).

In general, to generate random integers from **a** to **b**, use RND (a,b). For **a** and **b**, you can substitute any BASIC expressions.

WORLDS OF IF

And now we come to the statement that will enable us to write game-playing programs, the IF statement. Here is an example of an IF statement.

```
IF R=1 THEN PRINT "HEADS"
```

Well, if the value of R is 1, the computer will print the word HEADS on the screen. But, if R is not equal to 1, the computer will not print HEADS.

Below is a short program to simulate (imitate) coin flipping.

```
10 LET R=RND (0, 1)
20 IF R=1 THEN PRINT "HEADS"
30 IF R=0 THEN PRINT "TAILS"
40 GO TO 10
RUN

HEADS
HEADS
TAILS
HEADS
TAILS
TAILS
TAILS
HEADS
TAILS  We interrupted the computer here.
•  ← cursor
```

THE ESP MACHINE

How's your ESP? Here is a simple little program to check out your ESP. A discussion of the program follows the sample RUN.

```
100  PRINT "CAN YOU TELL WHAT I AM THINKING? LET'S FIND OUT."
110  PRINT "I WILL THINK OF A NUMBER FROM 1 TO 3 (1 or 2 or 3)."
120  PRINT "SO, YOU HAVE 1 CHANCE IN 3 OF GUESSING MY NUMBER."
130  PRINT
140  PRINT "GUESS MY NUMBER ... AND ... GOOD LUCK!!!"
150  LET X = RND (1, 3)                    X is the secret number.
160  INPUT "WHAT IS YOUR GUESS?"; G        G is the guess.
170  IF G = X THEN PRINT "THAT'S IT! MY NUMBER WAS"; X
180  IF G<>X THEN PRINT "AHA! I FOOLED YOU. MY NUMBER WAS"; X
190  GO TO 130
RUN

CAN YOU TELL WHAT I AM THINKING?  LET'S FIND OUT.
I WILL THINK OF A NUMBER FROM 1 TO 3 (1 OR 2 OR 3).
SO, YOU HAVE 1 CHANCE IN 3 OF GUESSING MY NUMBER.

GUESS MY NUMBER ... AND ... GOOD LUCK!!!
WHAT IS YOUR GUESS? 1
AHA!  I FOOLED YOU.  MY NUMBER WAS 3

GUESS MY NUMBER ... AND ... GOOD LUCK!!!
WHAT IS YOUR GUESS? 2
THAT'S IT!  MY NUMBER WAS 2
```

and so on. Play as many times as you like. Did you do better than 1 out of 3 . . . did you do much *better?*

Well, this program is pretty crude . . . only a beginning. We will explain how it works, then *you* can improve it. Next time, *we* will improve it!

Line 150 generates a random integer, 1 or 2 or 3, and assigns it as the value of X, the computer's secret number.

Line 160 gets the player's guess and then assigns it to the variable, G.

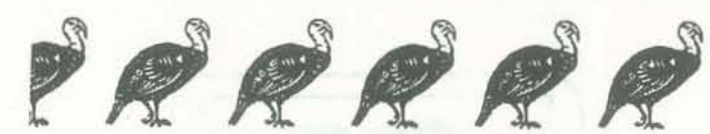
If G is equal to X, Line 170 will cause the computer to print THAT'S IT! MY NUMBER WAS followed by the value of X.

If G is not equal to X, Line 180 will cause the computer to print AHA! I FOOLED YOU. MY NUMBER WAS followed by the value of X. Note the symbol for "is not equal to."

<> means *is not equal to*

so, G <> X means *G is not equal to X*

NEXT TIME. More info about the IF statement, improvements to the ESP MACHINE, and more game-playing programs in Tiny BASIC.



DATA HANDLER: ANSWERS

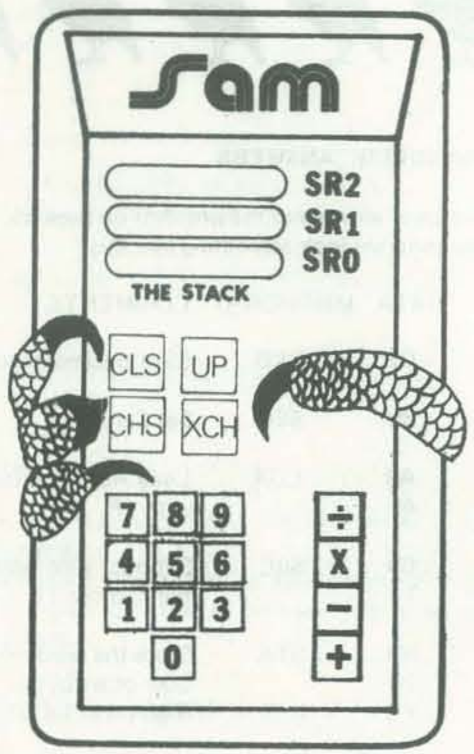
Here are typical answers to the problem on page 55. Does your program look something like this?

ADDR	DATA	MNEMONIC	COMMENTS
FC00	D8	CLD	Clear decimal mode.
FC01	38	SEC	Set Carry.
FC02 FC03	A9 4F	LDA	Load Accumulator with 4F.
FC04 FC05	E9 11	SBC	Subtract with borrow the number 11.
FC06 FC07 FC08	8D 20 FC	STA	Store the result in Low order byte High order byte.
FC09 FC0A FC0B	4C 09 FC	JMP	Jump to location FC09.

MAKE-BELIEVE COMPUTERS, Part Three

BY THE DRAGON

Our Dragon Emeritus continues to distribute his gems in this fourth part of an on-going series. Such Dragonese is suitable for beginning programmers and teachers of beginners.



This is the third of a series of articles to teach machine language and assembly language programming to beginners using a make-believe computer. The materials are intended for beginners or those who teach beginners. Are such articles of use to you? Let us hear from you soon if you want to see more of SAM.

In our November-December issue we introduced SAM, a handheld, make-believe computer. SAM has a *stack* of three *registers*. Each register can hold one number, which must be an *integer*. SAM also has a keyboard for entering numbers into the stack, moving them around in the stack, and performing arithmetic operations.

The plan was to proceed step-by step to add capabilities to SAM, with lots of sample programs at each step. So far, SAM has grown to the state you see pictured. Future expansions could include adding memory, GOTO, and IF statements. Gradually SAM would grow from an integer-based pocket calculator to an integer-based stack computer programmed in something that looks like an assembly language. But (alas), people are not filling our mailbox with solutions to the exercises for readers - the world is simply NOT beating a path to SAM's door. SO UNLESS WE HEAR FROM YOU, SAM WILL RETIRE AS OF THIS ISSUE.

For new readers, we recap how SAM works.

[CLS] Clears the stack to 0. Puts zero in each of the stack registers, SR0, SR1 and SR2.

[0] [1] . . . [9] Use these keys to enter numbers into SR0. You *cannot* key numbers directly in SR1 or SR2.

[UP] Pushes numbers in the stack up one place.
(1) Moves the number that was in SR1 into SR2,
(2) Copies the number that was in SR0 into SR1. This number is now in both SR0 and SR1.

[CHS] Changes the algebraic sign of the number in SR0. This operation changes a positive number to a negative number and vice versa; zero is not affected . . . it remains zero.

[XCH] Exchanges the numbers in SR0 and SR1. The number in SR2 is unaffected by this operation.

[+]
1) Adds the number in SR0 to the number in SR1 and puts the sum in SR0.
2) Copies the number that was in SR2 down into SR1. This number is now in both SR2 and SR1.

[−]
1) Subtracts the number in SR0 from the number in SR1 and puts the difference in SR0.
2) Copies the number that was in SR2 down into SR1. This number is now in both SR2 and SR1.

[X]
1) Multiplies the number in SR1 by the number in SR0 and puts the product in SR0.
2) Copies the number that was in SR2 down into SR1. This number is now in both SR2 and SR1.

[÷]
1) Divides the number in SR1 by the number in SR0 and puts the integer part of the quotient in SR0.
2) Copies the number that was in SR2 down into SR1. This number is now in both SR2 and SR1.

And a few notes:

If we key in a number immediately after pressing UP, the stack is *not* pushed. The number keyed in goes into SR0, but SR1 and SR2 are not changed.

If we key in a number immediately after pressing +, X or ÷, the stack is pushed. The number that was in SR1 is pushed into SR2 and the number that was in SR0 is pushed into SR1.

"INP a" means "key in the numerical value of a," and the number a may have more than one digit.

OK, on we go with our solutions. For new readers, the program is read from top to bottom. The final solution is in SR0.

(9) x^5 Two solutions.

(a)

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
X	x^3	x	x
X	x^4	x	x
X	x^5	x	x

(b)

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
UP	x^2	x^2	x
X	x^4	x	x
X	x^5	x	x

(10) In this problem, we used a instead of x to represent a number. In the above programs everyplace you see x, change it to a, and you've computed a^5.

(11) c^6 Three different programs.

(a) $c^6 = c \bullet c \bullet c \bullet c \bullet c \bullet c$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP c	c	0	0
UP	c	c	0
UP	c	c	c
X	c^2	c	c
X	c^3	c	c
X	c^4	c	c
X	c^5	c	c
X	c^6	c	c

(b) $c^6 = c^2 \bullet c^2 \bullet c^2$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP c	c	0	0
UP	c	c	0
X	c^2	0	0
UP	c^2	c^2	0
UP	c^2	c^2	c^2
X	c^4	c^2	c^2
X	c^6	c^2	c^2

(c) $c^6 = c^3 \bullet c^3$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP c	c	0	0
UP	c	c	0
UP	c	c	c
X	c^2	c	c
X	c^3	c	c
UP	c^3	c^3	c
X	c^6	c	c

(12) b^7 One solution, $b^7 = b^3 \bullet b^3 \bullet b$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP b	b	0	0
UP	b	b	0
UP	b	b	b
X	b^2	b	b
X	b^3	b	b
UP	b^3	b^3	b
X	b^6	b	b
X	b^7	b	b

(13) $m^8 = (m^4)^2 = (m^2 \bullet m^2)^2$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP m	m	0	0
UP	m	m	0
X	m^2	0	0
UP	m^2	m^2	0
X	m^4	0	0
UP	m^4	m^4	0
X	m^8	0	0

(14) x^{64} There are many, many ways to do this one! We will show you our favorite method.

PROGRAM	SR0	SR1	SR2	REMARKS
CLS	0	0	0	
INP x	x	0	0	
UP	x	x	0	
X	x^2	0	0	$x^2 = x \bullet x$
UP	x^2	x^2	0	
X	x^4	0	0	$x^4 = x^2 \bullet x^2$
UP	x^4	x^4	0	
X	x^8	0	0	$x^8 = x^4 \bullet x^4$
UP	x^8	x^8	0	
X	x^{16}	0	0	$x^{16} = x^8 \bullet x^8$
UP	x^{16}	x^{16}	0	
X	x^{32}	0	0	$x^{32} = x^{16} \bullet x^{16}$
UP	x^{32}	x^{32}	0	
X	x^{64}	0	0	$x^{64} = x^{32} \bullet x^{32}$

(15) $x^2 + x$ Only *one* INP. Easy, if we remember $x^2 + x = x + x^2$.

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
+	$x+x^2$	x	x

(16) $x - x^2$ Remember: the — key causes SAM to subtract the number in SR0 from the number in SR1.

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
—	$x-x^2$	x	x

(17) $a^3 + 2a$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP a	a	0	0
UP	a	a	0
UP	a	a	a
X	a^2	a	a
X	a^3	a	a
+	a^3+a	a	a
+	a^3+2a	a	a

(18) $2k^2 + 3k$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP k	k	0	0
UP	k	k	0
UP	k	k	k
X	k^2	k	k
UP	k^2	k^2	k
+	$2k^2$	k	k
+	$2k^2+k$	k	k
+	$2k^2+2k$	k	k
+	$2k^2+3k$	k	k

(19) $s^4 + s^2$

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP s	s	0	0
UP	s	s	0
X	s^2	0	0
UP	s^2	s^2	0
UP	s^2	s^2	s^2
X	s^4	s^2	s^2
+	s^4+s^2	s^2	s^2

(20) $(r^2 - r)^2$ This one is tricky. We take advantage of the fact that $(r^2 - r)^2 = (r - r^2)^2$ Why is this true?

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP r	r	0	0
UP	r	r	0
UP	r	r	r
X	r^2	r	r
—	$r-r^2$	r	r
UP	$r-r^2$	$r-r^2$	r
X	$(r-r^2)^2$	r	r

(21) $x^2 - x$ Well now what! In problem 16 we found $x - x^2$, but $x - x^2 \neq x^2 - x$. Here's where we need the CHS and XCH keys. We offer 2 solutions.

(a) CHS causes SAM to change the algebraic sign of the number in SR0. It has the effect of multiplying by -1. So $-(x-x^2) = x^2 - x$.

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
—	$x-x^2$	x	x
CHS	x^2-x	x	x

(b) XCH causes SAM to exchange the numbers in SR0 and SR1. In this program, we use XCH to put x^2 and x in the proper places before subtracting.

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
XCH	x	x^2	x
—	x^2-x	x	x

(22) $x^3 - x^2$ We cleverly note that $x^3 - x^2 = x(x^2 - x)$.

Remember, – subtracts number in SR0 *from* number in SR1.

(a)

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
—	$x-x^2$	x	x
CHS	x^2-x	x	x
X	x^3-x^2	x	x

(b)

PROGRAM	SR0	SR1	SR2
CLS	0	0	0
INP x	x	0	0
UP	x	x	0
UP	x	x	x
X	x^2	x	x
XCH	x	x^2	x
—	x^2-x	x	x
X	x^3-x^2	x	x

THE END (PROBABLY . . .
BUT SEE THE NEXT COLUMN)

LETTER TO THE DRAGON

Some of the problems for SAM in the March-April issue were interesting. For example, problems (21) and (22) *can* be solved with only one INP and without the two new keys. The solution for (22) is shown below; leave off the last X for the solution to (21). Note that if SR1 and SR2 contain the same number, the sequence +, – gives the same result as CHS.

I also found ways to write a program to raise a number to any power within SAM's capability, with reasonable efficiency and only one INP. For example, two ways to calculate x^{63} are shown below.

Method 1 works for any positive integer n. Fill in the program starting at the *last* line. Do the following until n is 1:

(A) If n is odd, write X and subtract 1 from n.
(B) If n is even, write X, write UP on the next higher line, and divide n by 2.

When n is 1, add another UP unless the initial n was 2. If the number to be raised to a power is not already in SR0, write INP x (or a, b, c, etc.), and write CLS above it.

Method 2 works only for integers that can be factored. For example, 63 is 7 times 3 times 3. Again filling in from the bottom, let n be one of these numbers (here 3), and use method 1; do not add INP and CLS. Then let n be another one of the numbers and use method 1 again. Repeat for the rest of the numbers, if any. Now add INP x and CLS.

I believe that the shortest program for raising a number to a given power can be found as follows:

Try method 1. If method 2 can be used, try it for each possible set of numbers to be multiplied. For x^{63}, the sets are (3, 21), (7, 9), and (7, 3, 3), so try method 2 three times. Choose the shortest program.

(22) $x^3 - x^2$

```
CLS
INP X
UP
UP
X
–
+
–
X
```

x^{63}: two methods

```
CLS        CLS
INP X      INP X
UP         UP
UP         UP
X          X
X          X
UP         UP
X          X
X          X
UP         UP
X          UP
X          X
UP         X
X          UP
X          UP
UP         X
X          X
X
```

Robert F. Miles
242 Abingdon Road NW
Lenoir, NC 28645

THE DRAGON RESPONDS

Aha! THIS is what we are looking for! More! More! Note the next to last step in (22). $2x - x^2$ is in SR0; x is in SR1 and SR2. We use –; for SRO

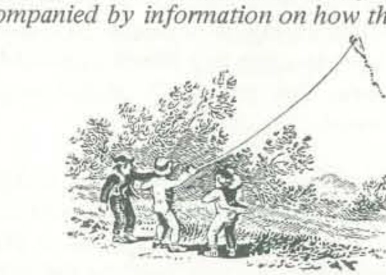
$$x - (2x - x^2) = x - 2x + x^2$$
$$= -x + x^2$$
$$= x^2 - x$$

and x is still in SR1 and SR2. Now X gives $x^3 - x^2$ in SR0!

PILOT

BY PHYLLIS COLE

PILOT is an easy-to-learn programming language especially suited for non-math-oriented computer programs. Kids as young as 5 or 6 have learned to program in PILOT, and so can you! Our last 3 issues carry articles on PILOT; we'll continue to publish information on PILOT and annotated programs – i.e. programs accompanied by information on how they work.



BACKGROUND

Last issue we published an experimental version of a PILOT interpreter written in the Z-80 assembly language written by our long-time friend Dean Brown of Zilog. Dean has now made a number of additions to his PILOT, and we'll show here how to use them. Once we're satisfied that all the bugs are out and desirable features in, we'll add paper tapes of Z-80 PILOT to the cheap mail order supply at Community Computer Center, 1919 Menalto Ave., Menlo Park CA 94025.

Dean does not call his version of PILOT "Tiny" though it certainly is that – even with the latest additions it takes only 700 bytes in assembly code. Z-80 PILOT is small and powerful and quite specialized at this point, since arithmetic isn't possible. PILOTs written in BASIC often use BASIC for computational purposes, but for now we're focussing on an easy-to-learn language that provides a way for beginners to write fun, interesting programs from the start – with no arithmetic expressions to scare off the math-hating 6 year old or math-shy novelist. See "Goldilocks" in the last issue as an example of such a program.

Let's begin with a brief review of some PILOT commands

R: the Remark command indicates that the text which follows on the same line is a comment about what that part of the program does, and is not part of the program.
T: the Type command types out on the terminal the text which follows on the same line
A: the Ask command causes the computer to pause and wait for user input

M: the Match command checks to see if what the user typed in for the last Ask command matches any of the items listed after the M: command. If there is a match, the match flag is turned on; if there is no match, the match flag is not turned on.

J: the Jump command causes a branch to the label named: that command will be executed next.

Y: and N: the Yes and No commands test the match flag. 'JY:' means 'jump, if the match flag is yes, or turned on.' If the Y or N command occurs by itself, the 'Type' command is used. 'N:' means 'type this if the match flag is no, or turned off.'

E: the End command stops execution of that part of the program in which it occurs

NEW FEATURES FOR Z-80 PILOT

Now for a discussion of three new features. Just how they're used is illustrated in the listing of the game at the end of this article.

1) Sometimes a Match command takes more than one line: if the last character in a Match command is a comma, then the next line of text is treated as part of the Match command

2) There is now a Use command, U:, which calls subroutines. A subroutine is used when the same sequence of instructions is needed several times in a program. Instead of typing them over and over you can type them once, give them a name, then 'Use' them as often as you need to.

   If you say 'U: RULE' then the computer next does the chunk of code labeled RULE; when it reaches an End command the subroutine is completed and the next command to be executed is the one *following* the statement 'U: RULE' – i.e. the program picks up where it left off when the subroutine was called.

3) There is now a Compute command, C:, for Z-80 PILOT which enables certain limited types of highly specialized arithmetic so that you can keep track of scores, number of guesses, etc. There are exactly 4 numeric variables available: I, J, K, L. *Note:* don't get confused – there's a J (Jump) command and also a numeric variable called J!

   To use the numeric variables, first type C:. Here are the limits as to what you can do with these variables.

   C: ZI The letter Z before a numeric variable sets the value to 0; so I is now 0.

   C: K A numeric variable *not* preceded by Z causes 1 to be added to the value; so 1 is added to the value of K. This may look like a trivial command, but it allows us to keep track of correct answers, number of problems presented or whatever we wish.

THE INOUT GAME

This game is enjoyed by kids and adults alike. The object is to guess the rule the game uses to decide whether a word is 'in' or 'out.' As a parlor game, a leader who knows the rule asks players who *don't* know the rule to take turns naming words. The leader tells the player whether the named word is 'in' or 'out.' If a player thinks she or he knows the rule, then the game reverses, as the leader gives a word and the player tries to correctly identify whether the word is 'in' or 'out.'

My object in writing this program was to teach myself PILOT, and to use as many different commands as possible. I originally programmed the game for PILOT 73 as implemented by Sylvan Rubin in BASIC on a PDP-11 at Stanford Research Institute. I've modified it somewhat to accommodate Z-80 PILOT.

Here's the general structure:

1) The player is given the first set of examples.
2) The player types in 5 words, one at a time, and is told whether each word is 'in' or 'out.'
3) The player is asked if she thinks she knows the rule. If she says 'yes' the computer types out 6 words, one at a time, and the player gets to tell whether each is in or out. If 4 or more are correct, the program assumes the rule is understood, congratulates the player, and the game ends.
4) If the player does *not* think she knows the rule she is offered more examples, if some are still available. If examples are desired, then the next set is printed out. Once the player has seen all 4 sets of examples, no more examples are offered.
5) If a player does not wish more examples, she is asked whether she wants to try some more. If she does, she gets to type in another 5 words, and the game goes back to steps 2, 3, etc. If not, the game ends.

Now we can test the value of the numeric variables by using =, (, and) . Parentheses are required as shown.


```
My name is Robin.
What's yours?
>Linda

OK, Linda, this game is called

          In and Out
Some things are in; some are out.
Try to figure out how it works.


Glass is in; plastic is out.
Eggs are in; bacon is out.
Jenny is in; John is out.
Massachusetts is in; California is out.


Now type a word.
I'll tell you if it's in or out.

  Word -
>game
 game is out.
```

```
  Word -
>hat
 hat is out.

  Word -
>babble
 babble is in.

  Word -
>ginger
 ginger is out.

  Word -
>balloon
 balloon is in.

Do you think you know the rule, Linda ?
>no

Do you want more examples?
>yes

Ribbon is in; tape is out.
Trees are in; bushes are out.
Bobby is in; Robert is out.
Grass is in; lawn is out.


Want to try some more?
>OK

  Word -
>baby
 baby is out.

  Word -
>stream
 stream is out.

  Word -
>entire
 entire is out.

  Word -
>look
 look is in.

  Word -
>silly
 silly is in.

Do you think you know the rule, Linda ?
>yes

OK, now tell me if each of these words is in or out.

River
>in
No, river is out.

Brook
>in
Good

Mirror
>in
Correct

Spoil
>in
No, spoil is out.

Spill
>in
Yes

Dancing
>out
Good

Hmmm, I'm not quite sure you've got it, Linda.

Do you want more examples?
>yes

Sweet is in; sour is out.
Boot is in; shoe is out.
Green is in; red is out.
Cheer is in; laughter is out.


Want to try some more?
>yes

  Word -
> yes
 yes is out.

  Word -
>no
 no is out.

  Word -
>grass
 grass is in.

  Word -
>lace
 lace is out.

  Word -
>lass
 lass is in.

Do you think you know the rule, Linda ?
>yes

OK, now tell me if each of these words is in or out.

River
>o
Right

Brook
>i
Good

Mirror
>i
Correct
```

```
Spoil
>i
No, spoil is out.

Spill
>i
Yes

Dancing
>o
Good

Great - looks like you've figured it out, Linda.

That's all for now, Linda.
```

Note that the program calls itself 'Robin,' a name that may be either male or female.

'*NA' is a label as the asterisk indicates. The player's input is stored in a variable of the same name starting with a backslash: \ NA. Wherever \ NA occurs in the program, the name typed by the player will be substituted.

'*AN' labels the player's input of a word; the word is stored in \ AN.

The 4-line match statement checks to see if double letters (either upper or lower case) appear in \ AN. If double letters occur the player is told \ AN is in; otherwise she's told \ AN is out. In both messages, \ AN is replaced by the word typed by the player.

Now 1 is added to K, the number of words typed. If K is less than 5, the program jumps to the label WORD and requests another word from the player. Otherwise the subroutine RULE is used.

If the player wants more examples, the program branches to TEST, where the set of examples that is presented (either EX2 or EX3 or EX4) depends on the value of I; the player gets each set of examples only once. After a set of examples is presented the program branches to QUES, where 1 is added to I. If the player does *not* want more examples, the program branches to QUES1.

```
R: Phyllis Cole  Z80   April 1977
T: My name is Robin.
T: What's yours?
*NA A:
R: \NA is player's name.
T:
T: OK, \NA, this game is called
T:
T:            In and Out
T: Some things are in; some are out.
T: Try to figure out how it works.
T:
T:
T: Glass is in; plastic is out.
T: Eggs are in; bacon is out.
T: Jenny is in; John is out.
T: Massachusetts is in; California is out.
T:
T:
T: Now type a word.
T: I'll tell you if it's in or out.
C: I
R: I = number of example sets given; now I = 1.
*ST C: ZK
R: K = number of words typed.
*WORD T:
T:  Word -
*AN A:
R: \AN is the word the player types.
M: aa,bb,cc,dd,ee,ff,gg,hh,ii,jj,kk,ll,
   AA,BB,CC,DD,EE,FF,GG,HH,II,JJ,KK,LL,
   mm,nn,oo,pp,qq,rr,ss,tt,uu,vv,ww,xx,yy,zz,
   MM,NN,OO,PP,QQ,RR,SS,TT,UU,VV,WW,XX,YY,ZZ
Y: \AN is in.
N: \AN is out.
C: K
J (K<5): WORD
U: RULE
J (L>4): FIN
R: If rule guessed, game ends.
J (I>3): FIN
R: If all examples shown, game ends.
*EXAM T:
T: Do you want more examples?
A:
M: Y,y,SU,su,Su,ok,OK
J Y: TEST
M: No,no,NO
J Y: QUES1
N: Please answer yes or no.
J: EXAM
*TEST J (I=1): EX2
J (I=2): EX3
J (I=3): EX4
*EX2 T:
T: Ribbon is in; tape is out.
T: Trees are in; bushes are out.
T: Bobby is in; Robert is out.
T: Grass is in; lawn is out.
T:
J: QUES
*EX3 T:
T: Sweet is in; sour is out.
T: Boot is in; shoe is out.
T: Green is in; red is out.
T: Cheer is in; laughter is out.
T:
J: QUES
*EX4 T:
T: Good is in; bad is out.
T: Marriage is in; divorce is out.
```


If the player wants to try some more she is branched back to ST, where the sequence of the player inputting 5 words begins again. Otherwise, the game ends.

The subroutine RULE asks if the player knows the rule. If the answer is no, the program branches to the end of the subroutine. If the answer is yes, 6 words are presented and the player must decide whether each is 'in' or 'out.'

Are you using PILOT? Can your interpreter be made cheaply available for others who wish to use PILOT? Have you PILOT programs to share? We're waiting to hear from you!

NO. 9 If at least 5 of the 6 answers are correct (L > 4) then the program assumes the rule is understood; the subroutine ends and then the game ends. If fewer than 5 answers are correct (L < 5) then the program assumes the rule is not understood; the subroutine ends and, if more examples are available, the game continues.

```
T: Aardvarks are in; anteaters are out.
T: Buttercups are in; roses are out.
T:
*QUES C: I
*QUES1 T:
T: Want to try some more?
A:
M: Y,y,SU,Su,su,ok,OK
J Y: ST
*FIN T:
T: That's all for now, \NA.
T:
E:
*RULE T:
T: Do you think you know the rule, \NA ?
A:
M: Y,y,SU,Su,su,OK,ok,,Gue,gue
J N: LAST
C: ZL
R: L = number of correct responses in RULE subroutine.
T:
T: OK, now tell me if each of these words is in or out.
T:
T: River
A:
M: O,o
Y: Right
C Y: L
R: Add 1 to L if answer is correct.
N: No, river is out.
T:
T: Brook
A:
M: I,i
Y: Good
C Y: L
N: No, brook is in.
T:
T: Mirror
A:
M: I,i
Y: Correct
C Y: L
N: No, mirror is in.
T:
T: Spoil
A:
M: O,o
Y: Right
C Y: L
N: No, spoil is out.
T:
T: Spill
A:
M: I,i
Y: Yes
C Y: L
N: No, spill is in.
T:
T: Dancing
A:
M: O,o
Y: Good
C Y: L
N: No, dancing is out.
T:
T (L>4): Great - looks like you've figured it out, \NA.
T (L<5): Hmmm, I'm not quite sure you've got it, \NA.
*LAST E:
```

BASIC Mortgages

BY TIM BARRY

Here in the San Francisco Bay area, as everywhere else these days, the price of housing compared to the average person's income has begun an almost unbelievable rise. As income and cost become separated by increasingly wider gaps, it becomes critical that you know what you can afford. Good housing is hard to find, so there is no point in wasting your time on places for which you will obviously not qualify. Here is a little BASIC program that can give you some help. It inputs all of the pertinent information and then gives you the bad news.

Your long term relationship to the bank is determined by several factors:

C = The cost of the house.

D = The amount of your down payment.

P = The principal financed (C-D).

I = The annual percentage rate of interest.

T = The number of years in the mortgage term.

These terms are run through a few simple calculations to produce

M = Your monthly mortgage payment.

The normal home loan is what is referred to as a 'fully amortized' loan. This means that you make a fixed number of constant payments, and when you make the last payment you own the home. This contrasts with some other types of loans where you make smaller payments and complete the loan with a large balloon payment.

With the fully amortized loan your fixed payment is apportioned between the loan principal and the interest owed. Over the life of the loan the proportion shifts. Initially, most of your money is being applied to the interest on the debt. As the term progresses, however, a larger and larger share of the money gets applied to the principal.

Without going into great, grisly detail, the monthly payment is computed as a combination of the normal compound interest equation and the normal annuity equation:

$$\text{Compound Interest: } A = P * (1 + i)^n$$

$$\text{Annuity: } V = A * (i/[(1 + i)^n - 1])$$

where

P = the initial amount borrowed

A = the total amount owed

V = the annual payment of the annuity

i = the annual interest rate

n = the number of years

Plugging in our already defined terms and adjusting the equations to be monthly rather than yearly, we find

$$A = P * (1 + I/12)^{T * 12}$$

$$M = A * [(I/12)/((1 + I/12)^{T * 12} - 1)]$$

Combining some terms, we now find that our monthly payment becomes:

$$J = I/12, S = (1 + J)^{T * 12}$$

$$M = (P * J * S)/(S - 1)$$

These equations can be readily converted to a BASIC program.

The program prompts you to enter the home cost, down payment, interest rate, and mortgage term. Once all the data is in it performs its calculations and prints out a table of the results. You can repeat the computation with different data to see what effect different conditions have on your payments. In particular, you will see that small changes in interest rate can have very noticable effects on the payments.

The program itself is written in IBM BASIC, but it should be easily adaptable to any of the popular mini and microcomputer BASICs.

One final note: The amount of payments shown by the program is strictly for the monthly mortgage. To this you must also add your property taxes and the insurance the lender will require you to carry over the life of the mortgage. Depending upon where you live, these will probably add another 10-20 percent to your monthly bill.

```
COST ØF HØUSE? >65000
YØUR DØWN PAYMENT? >10000
ANNUAL PERCENTAGE RATE? >8.5
MØRTGAGE TERM? >30

 HØME CØST          65000
-DØWN PAYMENT       10000
-------------------------
 AMØUNT FINANCED    55000

INTEREST RATE         8.5
MØRTGAGE TERM          30

MØNTHLY PAYMENTS   422.91

AGAIN? >YES
CØST ØF HØUSE? >65000
YØUR DØWN PAYMENT? >13000
ANNUAL PERCENTAGE RATE? >9.0
MØRTGAGE TERM? >30

 HØME CØST          65000
-DØWN PAYMENT       13000
-------------------------
 AMØUNT FINANCED    52000

INTEREST RATE          9.
MØRTGAGE TERM          30

MØNTHLY PAYMENTS   418.41

AGAIN? >NØ
```

```
10 REM BASIC MØRTGAGE CALCULATØR PRØGRAM
20 REM
30 REM    INPUT THE DATA
40 REM
50 PRINT 'CØST ØF HØUSE';
53 INPUT C
60 PRINT 'YØUR DØWN PAYMENT';
63 INPUT D
70 REM
80 REM PRINCIPAL = CØST - DØWN PAYMENT
90 REM
100 LET P=C-D
110 REM
120 PRINT 'ANNUAL PERCENTAGE RATE';
123 INPUT I
130 REM
133 REM CØNVERT PERCENT TØ DECIMAL, I = I/100
140 REM MØNTHLY PERCENT = I/12
150 REM
160 LET I=I/100
170 LET J=I/12
180 REM
190 PRINT 'MØRTGAGE TERM';
193 INPUT T
200 REM
210 REM  DATA IN. PERFØRM CALCULATIØNS
220 REM
230 LET S=(1+J)**(T*12)
240 LET M=(P*J*S)/(S-1)
243 LET M=(INT(M*100))/100
250 REM
260 REM  ØUTPUT THE RESULTS
270 REM
273 PRINT
275 PRINT
277 PRINT ' HØME CØST         ';C
278 PRINT '-DØWN PAYMENT      ';D
279 PRINT '-------------------------'
280 PRINT ' AMØUNT FINANCED ';P
283 PRINT
285 PRINT 'INTEREST RATE        ';I*100
287 PRINT 'MØRTGAGE TERM         ';T
288 PRINT
290 PRINT 'MØNTHLY PAYMENTS ';M
300 PRINT
310 PRINT
320 PRINT 'AGAIN';
323 INPUT T$
330 IF T$='YES' THEN 10
340 END
```

EXAGON

BY MAC OGLESBY

Mac says writing good rules is the hardest part of writing good game programs. For EXAGON, he neatly side-stepped the problem. Can anyone out there contribute short easy-to-understand EXAGON rules?

```
EXAGON

LEARN THIS GAME BY PLAYING IT...

    G . . . .
   W N . . . .
  . . I . . . Q
 . . . F . . S .
U . X . D . . . V
 . J P . . . . .
  . E L T . O .
   H R Y A . M
    B C K Z .

HOW MANY HUMAN PLAYERS  (1 OR 2)? 1
  OK, THE COMPUTER PLAYS THE #'S.
WHO GOES FIRST  (1=COMPUTER  2=YOU)? 1

THE #'S CHOOSE P

    G . . . .
   W N . . . .
  . . I . . . Q
 . . . # . . S .
U . # . D . . . V
 . J # . . . . .
  . # # T . O .
   # R # A . M
    B C # Z .

THE &'S CHOICE? .
PLEASE TYPE ONE OF THE LETTERS SHOWN? #
PLEASE TYPE ONE OF THE LETTERS SHOWN? G
THE #'S CHOOSE A

    & . . . .
   # & . . . .
  . . & . . . #
 . . . & . . # .
& . # . & . . . V
 . J # . . . . .
  . # # # . O .
   # R # # . M
    B C # # .

THE &'S CHOICE? R
THE #'S CHOOSE M

    & . . . .
   # & . . . .
  . . & . . . #
 . . . & . . # .
& . # . & . . . #
 . & # . . . . .
  . & & # . # .
   # & # # . #
    & & # # .

*** THE #'S HAVE 14 AND THE &'S HAVE 12 ***

TYPE  RUN  TO PLAY AGAIN...
```

```
EXAGON

LEARN THIS GAME BY PLAYING IT...

    . . D Y .
   . . J B . .
  . . . H . S U
 . . O . . . V .
. A . . . N . . R
 . . . Z . X . .
  . C K L . G .
   T W . . . Q
    E F I P M

HOW MANY HUMAN PLAYERS  (1 OR 2)? 1
  OK, THE COMPUTER PLAYS THE #'S.
WHO GOES FIRST  (1=COMPUTER  2=YOU)? 2

THE &'S CHOICE? E
THE #'S CHOOSE H

    . . D # .
   . . # # . .
  . . . # . S U
 . . O . . . V .
. A . . . # . . R
 . . . & . # . .
  . # & L . # .
   # & . . . #
    & F I P M

THE &'S CHOICE? L
THE #'S CHOOSE U

    . . D # .
   . . # # . .
  . . . # . & #
 . . & . . . # .
. A . . . & . . #
 . . . & . # . .
  . # & & . # .
   # & . . . #
    & & # & M

THE &'S CHOICE? M
THE #'S CHOOSE A

    . . D # .
   . . # # . .
  . . . # . & #
 . . & . . . # .
. # . . . & . . &
 . . . & . # . .
  . # & & . # .
   # # . . . &
    & # # & &

THE &'S CHOICE? D

    . . & # .
   . . & & . .
  . . . # . & #
 . . & . . . # .
. # . . . & . . &
 . . . & . # . .
  . # & & . # .
   # # . . . &
    & # # & &

*** THE #'S HAVE 12 AND THE &'S HAVE 14 ***

TYPE  RUN  TO PLAY AGAIN...
```


LETTER FROM MAC

I've heard that some readers have complained about "unnecessary" GOTO statements in my programs. In general, my programs *do* have GOTO's which might have been left out in a different programming style. Let me explain . . .

Until June 1975 my programs resembled tangled twine! They were poorly organized, uncommented and difficult to debug or understand, even by the programmer! Alas, the style is too common, and too often published.

The turning point for me as a programmer was the introduction on the Dartmouth system of Arthur Luehrmann's INDENT***. This editing program not only reformats a BASIC program, but also imposes a structural style. To clarify why my programs look as they do, I quote from KIEWIT COMMENTS, May 1975.

"A model example of 'standard' IF-THEN usage occurs when two different procedures are to be carried out, depending upon whether the stated condition is true or false. The following is a representative programming example:

```
10 IF A>0 THEN 40
20 LET B = 0
30 GO TO 50
40 LET B = A
50 (next statement)
```

Lines 20-30 can be thought of as the 'ELSE clause' and line 40 can be thought of as the 'THEN clause'. The meaning of such an IF-THEN structure becomes far clearer if the ELSE clause is indented:

```
10 IF A>0 THEN 40
20    LET B = 0
30    GO TO 50
40 LET B = A
50 (next statement)
```

Note that in reading an indented IF-THEN structure it is no longer necessary to read the line number referenced

```
100 ' NAME:  ELEMLIB***:EXAGON
110 '
120 ' BY:  MAC OGLESBY ON 07/12/76.
130 '
140 ' DESCRIPTION:  LEARN THIS GAME BY PLAYING IT.  FOR ONE OR TWO
150 '    PLAYERS.
160 '
170 ' INSTRUCTIONS:  TYPE "RUN" TO PLAY.
180 '
190 '
1000 DIM D(10,18)                           'D(,) STORES THE BOARD
1010 RANDOMIZE
1020
1030 LET L=26                               'WE START WITH 26 LETTERS
1040 LET P$(1)="#'S"                        'PLAYERS' SYMBOLS
1050 LET P$(2)="&'S"
1060 LET T=2                                'DEFAULT; THE #'S GO FIRST
1070
1080 FOR J=1 TO 4
1090    READ R(J),C(J)                      'CHANGES TO ROW, COL. TO LOOK
1100 NEXT J                                 '  DIAGONALLY UP AND DOWN
1110    DATA -1,-1,1,1,-1,1,1,-1
1120
1130 FOR J=1 TO 9
1140    FOR K=1 TO 17
1150       LET D(J,K)=32                    'ASC FOR SPACE
1160    NEXT K
1170 NEXT J
1180
1190 FOR J=1 TO 9
1200    FOR K=1+ABS(5-J) TO 17-ABS(5-J) STEP 2
1210       LET D(J,K)=46                    'DOTS FORM A HEXAGON
1220    NEXT K
1230 NEXT J
1240
1250 FOR J=1 TO 26
1260    LET R=1+INT(RND*9)
1270    LET C=1+INT(RND*17)
1280    IF D(R,C)=46 THEN 1300
1290       GOTO 1260
1300    LET D(R,C)=64+J                     'PUT 26 LETTERS ON BOARD AT RANDOM
1310 NEXT J
1320
1330 PRINT "LEARN THIS GAME BY PLAYING IT..."
1340 GOSUB 2470
1350 PRINT "HOW MANY HUMAN PLAYERS  (1 OR 2)";
1360 INPUT P
1370 IF P=2 THEN 1480
1380    IF P=1 THEN 1410
1390       PRINT "PLEASE TYPE 1 OR 2  ";
1400       GOTO 1360
1410    PRINT "  OK, THE COMPUTER PLAYS THE #'S."
1420    PRINT "WHO GOES FIRST  (1=COMPUTER  2=YOU)";
1430    INPUT F
1440    IF (1-F)*(2-F)=0 THEN 1470
1450       PRINT "PLEASE TYPE 1 OR 2  ";
1460       GOTO 1430
1470    LET T=3-F
1480 PRINT
1490
1500
1510 'MAIN MOVE LOOP
1520 LET T=3-T                              'T=1,2,1,2... OR, T=2,1,2,1...
1530 IF P=2 THEN 1900                       'SEE IF THERE ARE 2 PLAYERS,
1540    IF T=2 THEN 1900                    '  OR IF IT'S THE &'S TURN
1550
1560       'GENERATE COMPUTER'S MOVE
1570       LET E=0                          'RESET MOVE QUALITY INDICATOR
1580       FOR J=1 TO 9
1590          FOR K=1 TO 17
1600             IF D(J,K)<65 THEN 1820  'IGNORE NON-LETTERS
1610                LET Q=1              'RESET CHARACTER COUNTER
1620                FOR J9=1 TO 4        'CHECK DIAGONALS UP & DOWN,
1630                   LET R=J+R(J9)     'STARTING FROM EACH LETTER
1640                   LET C=K+C(J9)
1650                   IF D(R,C)<=32 THEN 1720  'CHECK FOR DIAGONAL'S END
1660                      IF D(R,C)=46 THEN 1690  'IGNORE DOTS
1670                         IF D(R,C)=35 THEN 1690  'IGNORE #'S
1680                            LET Q=Q+1  'TALLY POSSIBLE CAPTURES
1690                      LET R=R+R(J9)  'MOVE ALONG DIAGONAL
1700                      LET C=C+C(J9)
1710                      GOTO 1650
1720                NEXT J9
1730                IF Q>E THEN 1790     'LOOK FOR BEST MOVE
1740                   IF Q=E THEN 1760  'AS GOOD AS BEST PREVIOUS?
1750                      GOTO 1820      'INFERIOR MOVE; TRY NEXT LETTER
1760                   LET C9=C9+1       'TALLY POSSIBLE MOVES
```

by the IF statement. If the condition in line 10 is true, program control simply falls to the next unindented statement, line 40, a fact which makes indented programs far easier to follow. Note also that, should the ELSE clause itself contain nested within it another 'standard' IF-THEN structure, it can be further indented in the same way that nested FOR-NEXT loops can, as in this example:

```
10 IF A)0 THEN 70
20   IF A=0 THEN 50
30     LET B = 0
40     GO TO 80
50   LET B = ENDA1
60   GO TO 80
70 LET B = A
80 (next statement)
```

To summarize, 'standard' IF-THEN usage occurs when [a] the THEN clause comes after the IF statement [no backwards referencing] and [b] both the ELSE clause and the THEN clause are totally nested within any other BASIC structure that contains the IF statement, such as another ELSE clause or a FOR-NEXT loop. When this standard is followed, the logical flow of the program can be make remarkably transparent by adopting the above indentation style that reveals all properly nested structures."

All of my programs published by *People's Computers* can be shortened [and still RUN] by 1) omitting comments and spaces and 2) restructuring and/or restyling the BASIC code. However, the result may be a program which is harder to understand, especially after a lapse of time.

I expect that a reader who wants to use one of my programs on another system will:

1) modify the code as required by that system
2) shorten the program by dropping spaces and comments
3) insert new features and "improvements"
4) employ a satisfying and efficient programming style.

My programming style is still evolving and is certainly not the best or only way. But I think I'm pointed in the right direction, at least.

Best,

Mac Oglesby

```
1770                LET M(C9)=100*J+K  'STORE THIS MOVE
1780                GOTO 1820          'TRY ANOTHER LETTER
1790             LET C9=1              'RESET POSSIBLE MOVE COUNTER
1800             LET E=Q               'UPDATE QUALITY INDICATOR
1810             LET M(C9)=100*J+K     'STORE THIS MOVE
1820          NEXT K
1830       NEXT J
1840       LET E=M(1+INT(RND*C9))      'PICK AT RANDOM FROM BEST MOVES
1850       LET J1=INT(E/100)
1860       LET K1=E-100*J1
1870       PRINT "THE #'S CHOOSE ";CHR$(D(J1,K1))
1880       GOTO 2070                   'GO UPDATE BOARD
1890
1900 PRINT "THE ";P$(T);" CHOICE";
1910 LINPUT A$
1920 IF LEN(A$)<>1 THEN 2030
1930    CHANGE A$ TO A
1940    IF A(1)<97 THEN 1960
1950       LET A(1)=A(1)-32            'CHANGE LOWERCASE TO UPPER
1960    IF (90-A(1))*(A(1)-65)<0 THEN 2030  'LOOK FOR A LETTER
1970       FOR J1=1 TO 9
1980          FOR K1=1 TO 17
1990             IF D(J1,K1)<>A(1) THEN 2010
2000                GOTO 2070          'LETTER FOUND; GO UPDATE
2010          NEXT K1
2020       NEXT J1
2030 PRINT "PLEASE TYPE ONE OF THE LETTERS SHOWN";
2040 GOTO 1910
2050
2060 'UPDATE THE BOARD
2070 FOR J=1 TO 4
2080    LET R=J1+R(J)
2090    LET C=K1+C(J)
2100    IF D(R,C)<=32 THEN 2180        'NULL OR SPACE ENDS DIAGONAL
2110       IF D(R,C)=46 THEN 2150      'IGNORE DOTS
2120          IF D(R,C)<65 THEN 2140
2130             LET L=L-1             'THAT'S ONE LETTER LESS
2140          LET D(R,C)=32+T*3        'ASC FOR # OR &
2150       LET R=R+R(J)                'MOVE ALONG DIAGONAL
2160       LET C=C+C(J)
2170       GOTO 2100
2180 NEXT J
2190 LET D(J1,K1)=32+T*3               'UPDATE CHOSEN LETTER
2200 LET L=L-1
2210
2220 IF L=0 THEN 2300                  'ANY LETTERS LEFT?
2230    'IF 2 PLAYERS, PRINT BOARD AFTER EACH MOVE.
2240    'OTHERWISE, PRINT BOARD ONLY AFTER #'S MOVE.
2250    IF (P-2)*(T-1)<>0 THEN 2270
2260       GOSUB 2470                  'PRINT THE BOARD, THEN
2270    GOTO 1520                      '  CONTINUE THE GAME...
2280
2290
2300 GOSUB 2470                        'PRINT FINAL BOARD
2310 FOR J=1 TO 9
2320    FOR K=1 TO 17
2330       IF D(J,K)=35 THEN 2380
2340          IF D(J,K)=38 THEN 2360
2350             GOTO 2390
2360          LET H2=H2+1              'TALLY THE &'S
2370          GOTO 2390
2380       LET H1=H1+1                 'TALLY #'S
2390    NEXT K
2400 NEXT J
2410 PRINT "*** THE #'S HAVE";H1;"AND THE &'S HAVE";H2;"***"  'BELLS
2420 PRINT
2430 PRINT "TYPE  RUN  TO PLAY AGAIN..."
2440 STOP
2450
2460 'PRINT THE BOARD
2470 PRINT
2480 FOR J=1 TO 9
2490    FOR K=1 TO 17-ABS(5-J)
2500       PRINT CHR$(D(J,K));
2510    NEXT K
2520    PRINT
2530 NEXT J
2540 PRINT
2550 RETURN
2560
2570 END
```


▲ WOMEN & MATH PROJECTS: ▼
△ LAWRENCE HALL OF SCIENCE ▽

Three hundred and fifty girls, ages 6-14, have participated in eight-week courses called 'Math for Girls' at the Lawrence Hall of Science, a public science center on the Berkeley campus of the University of California. For many who enroll in this tuition course, math is a subject that either scares, mystifies, or bores them. Our intention is to introduce girls to 'hands-on' experiences in logical thinking and problem solving that stimulate their curiosity and interest in mathematics. Puzzles, games, and computer activities show a side of mathematics that can be as fun as it is challenging.

The need for such a class became apparent after we surveyed the enrollment of the Hall's classes in physical and life science, computer science, and mathematics. Female students comprised only 26% of the total number of students enrolled. Since Spring 1974, not only have 350 females taken Math for Girls, but the total number of females enrolled in other Lawrence Hall classes has doubled, increasing the total female enrollment to 36% of the student population.

Math for Girls is taught by women students at the University of California, Berkeley, who are majoring in mathematics, mathematics education, or computer science. The women are selected and trained on the basis of their interest and ability in mathematics, and their desire to act as role models of women in mathematics for their students. Throughout the eight weeks, time is set aside for discussion of girls' competency and interest in science and mathematics, and the stereotypic attitudes that can result in limited career expectations for women. The importance of electing science and mathematics courses in high school is stressed, since avoidance of such courses severely restricts an individual's choice of college major.

Activities in the eight-week sessions are selected to help students focus on fundamental concepts and relationships in mathematics: looking for numerical and geometric patterns; understanding variables and functions; strategically organizing information; classifying into sets; using coordinate systems and graphing; recognizing spatial relationships; estimating; and using logic to solve problems systematically. Work with manipulative materials is considered an important prerequisite to dealing with mathematical rules and principles on a symbolic level. Students work in groups, sharing different methods for solving problems, which helps them to see that mathematics is not a subject where each problem has only one method of solution.

This program has formed the basis for other projects designed to foster interest in areas where women are currently under-represented. In 1976, two conferences co-sponsored with Mills College and Alameda/Contra Costa Counties Mathematics Educators were held for 7th-12th grade women, their parents, and teachers. Reaching a total of 900 people, these conferences were held on the Mills College campus and provided female students with an opportunity to hear women scientists, mathematicians, engineers, and technicians describe their work and the problems and challenges associated with it; they participated in science and math workshops where they used materials for experimentation and investigation; and they met in small groups with 30 women representing a variety of scientific and technical careers. While the students were in these workshops, parents and teachers who accompanied them attended sessions on college admission and scholarship opportunities with representatives from 9 local and state colleges and universities; they spoke with women working in technical fields to gain an idea of the opportunities for women in non-traditional careers; and they attended a session on how to guide young women toward the widest range of future opportunities. Similar conferences have been held in 1977, and more are planned.

To increase the mathematical literacy of adults, the Lawrence Hall offers 'Math for Mathophobics,' an eight-week class designed to introduce people with math anxiety to the joy of mastering a few mathematical concepts and applications. The Hall is also developing workshops to assist teachers in providing mathematically interesting environments for their students, and alert them to the necessity of encouraging all students to pursue mathematics throughout high school.

◀◀ ◆ ▶▶

For more information, contact Nancy Kreinberg, Lawrence Hall of Science, University of California, Berkeley, California 94720: (415) 642-1823, or Rita Liff, Department of Mathematics and Computer Science, Mills College, Oakland, California 94613: (415) 632-2700 x308.

THE DATA HANDLER USER'S MANUAL: PART 3

by Don Inman

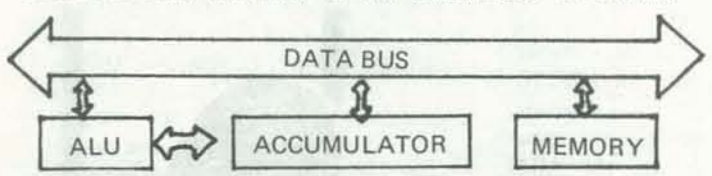
Don Inman is a teacher on sabbatical who's been working with teachers in the San Jose School District. Under Don's guidance, the teachers have built Data Handlers, complete microcomputer systems based on the 6502 microprocessor, and are now learning to use them. This is the third in a series of articles aimed at teaching relatively inexperienced people how to do assembly language programming for the 6502.

This user's manual is designed to serve both as a self-teaching guide and as an outline for a course at the beginning level of computer science. While it deals specifically with the Data Handler, it can easily be adapted to other microcomputers using the MOS Technology 6502.

The first semester course consists of nine two-hour class sessions, the first two of which were spent constructing the systems. Part 1 of our series covered session 3 of the course: system specifications, binary and hexadecimal notation, and how to do a preliminary checkout of the system. Part 2 of our series covered data transfer, and the use of a simple data transfer program. Part 3 covers session 5: the arithmetic logic unit.

SESSION V - THE ARITHMETIC LOGIC UNIT

Before we tackle our first practical problem, let's take a look at the Arithmetic Logic Unit (ALU). The eight-bit arithmetic unit interfaces to the accumulator as shown:



The ALU contains circuitry to perform addition of two 8-bit values and gives an 8-bit binary result plus a carry. An 8-bit binary number can represent decimal values from 0 through 255.

O O O O   O O O O = O

● ● ● ●   O O O O

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255

If the sum of two numbers is greater than 255, a carry bit is generated, and the excess over 256 is expressed in the eight-bit result. The carry bit is not a part of the data and is not used in transfer operations. However, we will learn how to test the carry bit to see if it was generated or not. If there had been a carry generation, you could use a Branch or Carry Set instruction to branch to a correction routine (the correction routine program would be created by the programmer to fit the purpose for the particular problem).

EXAMPLE WITH NO CARRY

Decimal		Hex
110	=	6E
+103	=	+67
-----		-----
213	=	D5

Answer is less than 256 so can be displayed as an eight bit number.

EXAMPLE WITH CARRY

Decimal		Hex
210	=	D2
+ 103	=	+ 67
-----		---
313	=	139 (100 + 39)

The result is greater than decimal 255. The 139 hex value cannot be displayed in 8-bits. A carry is set to represent the value 100 (256 decimal) and the 39 is expressed in the 8-bit result.

From earlier sessions we know how to load a number and store a number. We will need three new instructions to perform an 8-bit addition. The ADC instruction performs the addition of two numbers and the carry (if it exists). The CLC instruction clears the carry bit so that a previous carry will not be added when not desired. The CLD instruction is used to clear the decimal mode. The Data Handler can perform addition and subtraction in either binary mode or in a decimal mode (which will be discussed later). At the present time we will be working in the binary addition mode. The Data Handler when powered up (turned on) may come up in either mode, thus it is necessary to either CLEAR or SET the decimal mode when preparing for an addition or subtraction.

This sample program adds two hexadecimal numbers.

ADDR	DATA	MNEMONIC	COMMENTS
FC00	D8	CLD	Clear decimal mode if set.
FC01	18	CLC	Clear carry if set.
FC02	A9	LDA	Load accumulator immediate
FC03	11		First number to add.
FC04	69	ADC	Add to accumulator with carry
FC05	4F		Second Number
FC06	8D	STA	Store the result
FC07	20		Low order byte
FC08	FC		High order byte
FC09	4C	JMP	Jump to
FC0A	09		Low order byte
FC0B	FC		High order byte

Enter this program by the LOADING PROGRAMS procedure. Don't forget to load the initialization vector in FFFC and FFFD. Check your program for errors by the Examining Programs procedure and make any corrections. Then run the program. Your answer is found by examining location FC20. That location should show:

DATA 0●●0 0000 = 60 hex

ADDR ●●●● ●●00 00●0 0000 =FC20 hex

You should practice running this program several times with different values in locations FC03 and FC05. To change these values:

1. Press CL key.
2. Press AD key
3. Key in desired value (try the ones below)

FC03	FC05	Result
43	22	65
5C	55	B1
38	8E	C6
A6	5F	--

FC03	FC05	Result
45	A2	E7
70	3F	AF
44	77	--
B5	46	--

One large disadvantage to the above program is the fact that the numbers to be added are contained in the program itself. To make a change we must alter our original program. A more general approach would be to have the numbers stored in some memory location. We would then use the ABSOLUTE addressing mode rather than the IMMEDIATE mode used in this program. Although the advantage of this method is not immediately obvious, as our programs become more complex, it will become apparent.

In addition to the above change, let's modify the program to add three numbers rather than two. No new instructions will be needed.

Make up some of your own numbers, run the program, and check answers by hand calculation.

PROGRAM TO ADD THREE NUMBERS

ADDR	DATA	MNEMONIC	COMMENTS
FC00	D8	CLD	Clear decimal mode.
FC01	18	CLC	Clear carry.
FC02	AD	LDA	Load accumulator—absolute
FC03	00		Low order byte (first number)
FC04	FD		High order byte.
FC05	6D	ADC	Add with carry—absolute.
FC06	01		Low order byte (second number)
FC07	FD		High order byte.
FC08	6D	ADC	Add with carry—absolute.
FC09	02		Low order byte (third number)
FC0A	FD		High order byte.
FC0B	8D	STA	Store result—absolute
FC0C	10		Low order byte
FC0D	FD		High order byte.
FC0E	46	JMP	Jump to
FC0F	0E		Low order byte
FC10	FC		High order byte.

This now gives us a program which will add any three numbers that are stored in locations FD00, FD01, and FD02. If the three numbers are to be changed, our program remains the same. We merely change the Input data in the above locations. Our program is still only usable for three or fewer numbers. Suppose we wanted to add five numbers. We would have to rewrite our program.

Let's go back to the addition program to add two hexadecimal numbers. With some slight modifications, we can come up with a subtraction program. In the addition problem we cleared the carry bit at step FC01. Subtraction needs just the opposite at this point, Change that step to SEC (set carry flag). The instruction op code is 38. Step FC04 must be changed to SBC (subtact with borrow), op code E9. Since the number 11 used at step FC03 is smaller than the number 4F used at step FC05, those two should be reversed so that we subtract 11 from 4F. That's all there is to it.

Continued on page 55.

©OMPUTERS AND ©OPYRIGHT LAW

BY PHYLLIS COLE

The copyright laws now in effect were passed in 1909. But on January 1, 1978 new copyright laws go into effect. These new copyright laws are not yet completely drafted: the issue of just how the new laws apply to new technology has not been finalized.

When Congress passed the new copyright laws it created the Commission on New Technological Uses of Copyrighted Works, 'CONTU.' CONTU has been holding public hearings in order to study and compile data on

1) the reproduction and use of copyrighted works
   a) in conjunction with computers
   b) by various forms of machine reproduction
2) the creation of works by computer.

On December 31, 1977 (or August 1, 1978, if approval for an extension is received) CONTU is charged with reporting to Congress

such changes in copyright law or procedures that may be necessary to assure for such purposes access to copyrighted works and to provide recognition of the rights of copyright owners.

During the past 6 months, CONTU has heard testimony regarding use of copyright in the areas of photocopying, computer software, computer data bases, and computer-created new works.

Now CONTU is holding conferences to obtain input on the issues from consumer group leaders and other public interest advocates. I attended the first such conference in Washington, D.C., on May 2; I shall attend a second one on June 13. In my opinion, to date the commission has received inadequate input from the world of microcomputers. The world of micros differs from IBM-giant land, which manages quite well with trade secrecy agreements. Micro hardware and software developers aren't far from being cottage industries. Micro software may remain a cottage industry indefinitely, without any trade secrecy agreements to protect it. After all, how can a cottage industry operating a mail order house prevent purchasers from copying?

But probably copying isn't usually going to be a problem. Tom Pittman and others seem to have found a way around it so far by providing software so cheap that it won't be ripped off like Altair BASIC. What may be a problem is when a company or individual copies your software in order to *sell* it. Can copyright help in such a case? Maybe, if you have enough money to support a court fight (and against a company that may be hard for an individual) *and* if you have evidence to support your claim of prior development.

There are many issues to be considered. For example, what if only individuals (not corporations) were allowed to hold copyrights? Would it be preferable to rewrite the patent laws to include software instead of leaving software in the copyright domain? Under patent laws a greater and possibly excessive amount of control over the software would be permitted than under copyright law. Whatever line of thought you pursue, consider the effect on author, vendor, and consumer. Who benefits most?

So much for my ruminations: I urge you to send your comments to CONTU as to the proper role of copyright laws and procedures. Assume that your remarks will be directed to persons who are relatively unfamiliar with computer terms: define as you go.

Possibly CONTU will hold public hearings in various parts of the country later in the year. If you are interested in testifying at such a hearing, let CONTU know. Send comments to

Arthur J. Levine, Executive Director
CONTU
Crystal Mall No. 2
Arlington, VA 22202

CONTU is also anxious to gather more input from consumers and public interest advicates. Since PISA is preparing a position for CONTU, such comments should be directed to Mr. Levine and also to

Andy Horowitz
Public Interest Satellite Association
55 West 44th Street
New York, NY 10036

CONTU'S MISSION

CONTU is considering what changes are needed in copyright laws and procedures with respect to:

1) the reproduction and use of copyrighted works of authorship
   a) in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information, and
   b) by various forms of machine reproduction, not including reproduction by or at the request of instructors for use in face-to-face teaching activities; and

2) the creation of new works by the application or intervention of such automatic systems of machine reproduction.

CONTU'S QUESTIONS

CONTU Commissioners are seeking answers to questions in all the above areas. Among the software-related questions are:

1) Should a computer program be copyrightable? – patentable? – or both? – should the type of protection afforded vary according to the nature of the program? – for what length of time should protection be available?

2) Should copyright protection of computer software be limited to the right to make and vend copies of the program or should the right extend to the use of a program to operate a computer in a manner similar to the performance right in a musical or dramatic work?

3) What constitutes copying of a computer program? – making a new version in a similar medium? – inputting the program into memory for execution?

4) What type of additional legal protection for software is needed, as distinguished from more effective enforcement of the present law?

5) How can additional protection for software be granted in such a way that it does not lead to a monopolization of the basic ideas and structure upon which the particular program is based?

6) Would stronger copyright protection for software encourage increased sale of proprietary software products? – would it lessen reliance on restrictive licensing arrangements based on trade secrecy?

7) How should the copyright notice be affixed to the software product?

8) In what form should registration copies of programs be deposited with the Library of Congress? – listings? – tapes? – flow-charts? – complete documentation packages? – some form comparable to dated and witnessed lab notebooks?

9) How would the changes suggested affect the proprietors and users of software products?

REFERENCES

Reference material includes these government documents:

- Public Law 94-553, The Copyright Act of 1976 (enactment of S.22, Senate Bill 22 of the 94th Congress)
- The Conference Report accompanying S.22
- The House of Representatives Report accompanying S.22
- The Senate Report accompanying S.22.

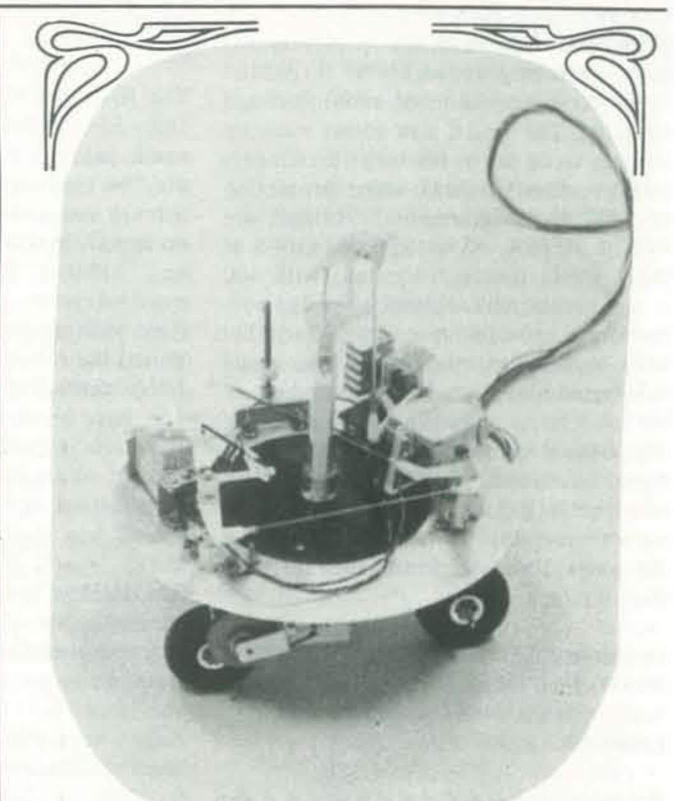
DATA HANDLER, continued from page 53.

Complete this program to subtract two numbers. Sample answers are on page 35.

ADDR	DATA	MNEMONIC	COMMENTS
FC00			
FC01			
FC02			
FC03			
FC04			
FC05			
FC06			
FC07			
FC08			
FC09			
FC0A			
FC0B			

Try your program with various numbers. See what happens if the subtrahend is greater than the minuend (bottom number bigger than top). Can you interpret the results? Read the section on signed arithmetic in the MOS Technology Programming Manual.

In the next session we'll learn to use loops and indexed addressing. With these new tools we can write more general programs which will fit a wide variety of needs.



TURTLE? WALDO? ROBOT?

Here's a close-up of Dr. Franz Frederick's friend Waldo, a computer-controlled turtle/robot. No, that's not a tail, it's the cord linking Waldo to a computer.

ANNOUNCEMENTS

HARDWARE

64K RAM BOARD

A fully tested 64K RAM board is being offered by Extensys Corp. of Sunnyvale CA. This board meets S-100 computer interface specifications, including Altair and Imsai units, and allows memory addition up to 1,048,576 bytes.

The 5x10 inch PC board contains 65,536 bytes and has hardware provision for bank switching to add over 1 million bytes--far beyond normal microprocessor capacity. The board also allows memory address to be set in 8K bytes increments and provides hardware-write protection in 16K byte increments. Voltages are +12 at 300mA, +5 at 750mA, and -5 at 1mA. Cycle time is 500 n sec, with 400 n sec access time. Memory overlap protection is provided to ensure no conflict with existing memories. All boards are fully assembled, tested, and burned in.

The 64K RAM board is $1,495. A 32K board is offered at $895, and a 48K board at $1,195.

To order direct, or for further information, contact:

Extensys Corp.
592 Weddell Drive, Suite 3
Sunnyvale CA 94086
(408) 734-1525

MEGABYTE MEMORY

IMSAI Manufacturing Corporation announces the world's first megabyte memory system for microcomputers. System modules offered at this time include: 65 K, 32K, and 16K Ram Boards controlled by IMSAI's unique Intelligent Memory Manager (IMM).

The IMSAI megabyte memory system is a complete memory system for the IMSAI 8080 and other S-100 bus computers. It consists of the RAM-16, RAM-32 and RAM-65 dynamic memory boards and intelligent memory manager (IMM) controller board. It may be implemented in a variety of configurations ranging from a single board 16K byte conventional memory to a 17-board one megabyte system with an Intelligent Memory Manager/interrupt controller.Larger multiprocessor systems may be implemented by using multiple IMSAI mainframes and the IMSAI Shared Memory Facility. Shared memory blocks can be up to 65K bytes and each processor can address up to one megabyte total of shared and local memory.

The RAM-16, RAM-32 and RAM-65 are 16K, 32K and 65K byte low power dynamic memory boards respectively. They may be used alone or in combination to form a conventional memory system of up to 65K bytes. With 400 ns access time and "hidden refresh", no wait states are required when accessed by the MPU. (One wait state may occasionally be required for refresh when accessed by some DMA controllers). The address of each 16K byte block is individually selectable. Provision is included for read and write-protect and expansion to one megabyte when used with the IMM controller board.

The IMM is an Intelligent Memory Manager/interrupt controller board. It provides for memory expansion to one megabyte, write protect for each 1K block in the extended Space, read protection, fully vectored interrupts, "time of day" clock and real time clock.

The IMM Firmware features are tailored to facilitate implementation of data acquisition, realtime processing, logging, time sharing and other systems which are likely to require very large memories. If desired, the user can specialize his system by selecting an added cost 8755 option and modifying the program.

The memory modules are offered in both assembled and kit form. Prices are as follows:

65K Ram Board Kit -	$2,599
Assembled -	$3,899
32K Ram Board Kit -	$ 749
Assembled -	$1,099
16K Ram Board Kit -	$ 449
Assembled -	$ 679
IMM ROM Control Kit -	$ 299
Assembled -	$ 399
IMM EROM Control Kit -	$ 499
Assembled -	$ 699

For more information, contract:

Michael Stone
IMSAI Manufacturing Corporation
14860 Wicks Boulevard
San Leandro CA 94577
(415) 483-2093

EAST COAST FLOPPIES

Innovex Corporation Today announced its new Series 400 diskette drive, following an extensive market study and development effort. Offering an array of new features that were previously unavailable in the market, the Series 400 marks an important advance in the state of the art in floppy disk technology. Unique new features include automatic head-unload and stepper motor time-outs, bi-directional failure detector, 6 different L.E.D. activity indicators, and 50-pin ribbon cable or twisted pair interfacing compatibility.

Both the Model 410 (soft-sectored, IBM-compatible) and the Model 420 (hard-sectored) provide single and double density recording capability while incorporating electrical features previously unavailable on any other floppy disk. In addition, a proprietary data separator design, coupled with a digital noise filter and a unique way of handling recorded signals, results in 35 percent greater data integrity margins than available from the closest competitor.

Prices for the Innovex Series 400 range from $575 in single quantities to $435 each for orders of 100. Deliveries began in March and are quoted at 30 days ARO. Innovex Corporation, the only East Coast manufacturer of floppy disks, is located at 75 Wiggins Avenue, Bedford, Massachusetts.

MICRO-FLOPPY SYSTEM (FORTRAN IV AVAILABLE)

Electronic Product Associates has announced a complete Micro-Computer/Floppy Disk System for the 6800. Both the Floppy Disk and Micro-Computer are mounted in identical, ruggedized, medium blue aluminum cabinets. The Micro-68b computer lists for $1878.00 and comes complete with 8K words of RAM memory, 1K MIK-BUG monitor system and audio Cassette, TTY, RS232C interfaces. The Micro-68b has a 13 slot exorcisor compatible motherboard and a 20 amp 5 volt power supply (± 12 volts at 1 amp also included). Power connections are provided for 115/230 volts 50/60 hz. Input/Output can be handled via either the built-in hexidecimal keyboard and LED display or via 20 ma. or a RS232C serial interfaces.

The IBM compatible Micro-68 Floppy Disk System is available in either single or dual configurations and comes complete with power supplies, interface to the Micro-68 computer controller and disk drive electronics. Each disk holds ¼ million bytes of information and is exorcisor compatible.

The Micro-68b lists for $1878.00; the Single Floppy Disk System for $2595; and the Dual Floppy System for $3295. Software available includes Fortran IV, Basic, Assembler, Editor and Floppy Disk Operating System. Delivery is one week. U.S. prices quoted.

For further information please contact:

Patti Neumann
Director of Marketing
Electronic Product Associates, Inc.
1157 Vega Street
San Diego, CA 92110
714-276-8911

LOW COST HARDWARE

Ithaca Audio offers the lowest prices in the U.S. on the IMSAI 8080 by depending on word of mouth advertising. Other offerings include quantity discounts, neat (and new) surplus, blank boards on anything Ithaca Audio designs, and time and advice to anyone who calls! If our price isn't the best, let us know.

IMSAI 8080 with 22 slot mother		$590.00
Seals 8K Kit 250ns		236.00
Blank RAM Boards 2102 type	8K	25.00
	4K	18.00

All Digital Group 12% off; quantity and club discounts.

For more information, contact Ithaca Audio:

Sales PO Box 91 Ithaca NY 14850
Office 410 College Ave, Ithaca
Phone 607-273-3271

Z-80 MICROPROCESSOR BOARD

A single board Z-80 microprocessor system has been announced by Mini Micro Mart. It is software compatible with the Intel SBC 80/10 Board but uses the new Zilog Z-80 CPU. There is provision onboard for three 2708 E PROMS, 1K of static RAM, two 8255's (providing 9 parallel ports), and an 8251 USART for a serial interface. Both a 20 mil current loop TTY and an RS-232-C interface are on-board with provisions for baud rates from 110 to in excess of 9600.

Full address decoding is provided for both the on-board memory and the I/O devices. It will also run Altair, IMSAI, and Processor Technology software with very minor modifications. The board is 7" x 10½" of plated thru epoxy glass material and with gold fingers on a dual 43-.156 center format.

It is part of a complete family which includes 4K and 16K static RAM boards, PROM boards and a scientific calculator interface. A complete system in a self-contained table-top cabinet, including Teletype printer, is available. The CPU board system is available in kit form starting at $199.95, or assembled and tested at $249.95, in single unit quantities. As an introductory offer, listings for our 5K BASIC and a powerful operating monitor, which provides for entry and dumping in both Octal and Symbolic, are included.

For additional information, contact Mini Micro Mart, 1618 James Street, Syracuse, New York 13203, (315) 422-6666. Delivery is stock to 30 days.

IMSAI 8048 CONTROL COMPUTER

IMSAI Manufacturing Corporation is pleased to announce the IMSAI 8048 Control Computer. This complete control computer on an 8½" x 10" board is on Intel's new 8048 micro computer chip. It is the *World's First* Single Chip Control Computer that contains all of the following features:

1. 8 bit CPU.
2. 2.5 Microsecond instruction cycle, 96 instructions.
3. BCD arithmetic capability.
4. 1K words of ROM or compatible EPROM program memory.
5. 64 words of Internal register memory.
6. 27 I/O lines.
7. Internal Timer/Event counter.
8. Oscillator and Clock Driver.
9. Reset circuit.
10. Interrupt Circuit.
11. Uses Single 5-Volt Supply.
12. TTL Compatible.

In addition, to create a one board user programmable controller suitable for use with model railroads, energy conservation systems, ham radios, household appliances, lights, light shows, and a myriad of other applications, the following system features have been added:

1. Cassette Interface.
2. Serial I/O (RS232, current loop).
3. 5 relay capable of handling 2 Amps at 220 Volts, 3 Amps at 110 Volts, or 5 Amps at 24 VDC.
4. 1K (optional additional 1K) of user programmable program memory.
5. DC power supply or battery operated.

The IMSAI 8048 Control Computers are offered in both assembled and kit form. Prices are as follows:

IMSAI 8048 Control Computer (ROM version)
- $249 - Kit
- $299 - Assembled

IMSAI 8048 Control Computer (EROM version)
- $399 - Kit
- $499 - Assembled

Michael Stone
IMSAI Manufacturing Corporation
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093

6800 CASSETTE/TERMINAL INTERFACE

PerCom Data Company today announced off-the-shelf availability of the first high speed, self-clocking cassette/terminal interface for the Southwest Technical Products Corporation 6800 Microcomputer.

Designated the CIS-30+, the dual function unit interfaces cassette data at a user selectable 30, 60, or 120 bytes per second, and provides RS-232 interfacing at 300, 600, or 1200 baud.

The CIS-30+ users manual and optional test cassette include PerCom cassette operating software, however, the 6800-resident Mikbug is the only software needed except for loading at 120 bytes per second. The CIS-30+ also plays unmodified SWTPC cassette software.

The CIS-30+ kit is $69.95. Assembled, it costs $89.95. An instruction manual is included.

If not available at a local dealer, orders may be placed directly with PerCom Data Company, 4021 Windsor, Garland, Texas 75042 (214) 276-1968. Mail and telephone orders require an additional 5% for shipping, and 10% if COD. Texas residents must also include 5% sales tax.

SUPER VIDEO CHIP AVAILABLE

Jim Day reports that Standard Microsystems Corp., Hauppauge, NY 11787 (516/273-3100) recently announced a new *user-programmable* video timer-controller chip containing all of the logic for timing and formatting video displays. This 40-pin NMOS chip, designated the 5027, allows program control of such parameters as characters per line, lines per frame, raster scans per character, etc. Control of the chip is via seven 8-bit registers that can be loaded from a parallel I/O port, using four register address lines and a chip-select line to provide complete control by a microprocessor such as the 6800 or 8080. The 5027 will sell for about $26 in OEM quantities.

VIDEO GAME CHIP

A new game chip intended for the black and white TV market is now available from National Semiconductor Corporation. The MM5789 is an IC that has tennis, hockey and handball in the format similar to the MM57100, National's color video game chip.

"The key feature of the new low-cost black and white chip," said product marketing manager Gary Blondefield, "is that it is completely compatible with our color TV chip. Our intention is to make all our game chips interchangeable. A manufacturer can build different games by simply using any of our game chips with the same circuit board and system configuration."

The game chips may also be integrated directly into a TV set without going through the antenna, requiring only two chips instead of three. The MM5789 generates the composite video, and the MM53104 operates directly with a 3.58 MHz crystal. To accomplish this, the video signal is inserted into the television set at a point after the video detector and the audio are switched into the sound section of the set. In this way, an ordinary black and white TV set may be modified to include games by adding a multipole switch power supply, game chipset and components, and a push-button switch to select the game.

SOFTWARE

8080 TIMING INFO

Pragmatic Systems has introduced a new software design aid to help 8080 users compute execution times for speed critical program sections. Microcomputers are commonly used in applications where program execution speeds must be accurately known. Until now much of this analysis has been done by approximation or by trial and error. This new card helps simplify this analysis.

The Pragmatic Systems 8080 Instruction timing Reference Card presents instruction excution times and characteristics of all 8080 instructions in a concise and easy to use format. The general timing equations for the 8080 are included along with pre-computed execution time columns for 8080 systems with a 2 MHz clock and memory access times from 0 to 1500 ns.

One problem almost unique to microcomputer applications is timing analysis of systems with different speed memories. This often occurs when the programs are stored in ROM and data is stored in RAM. Failing to account for these speed differences can result in timing errors of over 40%. The card makes analysis of these systems easy by showing the number of accesses each instruction makes to both read only and read/write memory, and by providing pre-computed execution time columns for these split speed memory applications.

The back of the card contains instructions and equations for computing execution times of existing programs in any 8080 system. It also shows how to use the card to design program delay loops with specific execution times. Small sample programs are included as examples.

The card is printed on heavy stock and punched for three-ring bingers. It is ideal for use as a notebook reference.

The card price is $2.00 each or three for $5.00. All orders shipped postpaid from Mt. View, CA. (California residents should add 6.5% sales tax.) Volume discounts are available and dealer inquiries are invited. Pragmatic Systems, P.O. Box 43, Mt. View, CA 94042.

8080 FORTRAN IV COMPILER

Microsoft, an Albuquerque based software development firm, has announced the availability of a FORTRAN IV compiler for the 8080 microcomputer. Called FORTRAN-80, the initial release of this compiler is a full implementation of ANSI Standard Fortran with the exception of the double precision and complex data types.

FORTRAN-80 provides three data types including: logical (one byte), integer (two byte), and real (four byte floating point). An extended version of FORTRAN-80 with double precision and complex data types is forthcoming.

The compiler generates pure, relocatable code (may be placed in ROM), and the runtime package may also be placed in ROM. The one-pass compiler requires less than 12K bytes of memory, and the run-time system less than 6K bytes.

Individual copies of FORTRAN-80 may be purchased for $500 including documentation. Manual $15. OEM licenses available.

Contact: Paul Allen
Vice President
Microsoft
819 Two Park Central Tower
Albuquerque, N.M. 78108
(505) 256-3600

CO-RESIDENT 8080 EDITOR/ASSEMBLER

Tycon, Incorporated is pleased to announce its co-resident editor/assembler (TEA) for 8080 systems. Requiring only 5K of memory (R/W or PROM) it is completely I/O independent relying upon its own I/O software or the I/O routines already available in a user's system. The Tychon Editor/Assembler accepts both octal and hexadecimal values throughout the program and the program listings may be in either octal or hexadecimal form. The switch between octal and hex is made at any time using keyboard commands. The TEA package is the only editor/assembler available which allows the user to easily change the numbering system used. The editor/assembler is relocatable using a special relocator within the program which will place TEA anywhere in the 8080's memory space. The program is available in 1702A or 2708 PROMs and on paper tape. Listings are also available. Delivery is immediate. Prices start at $35 for a paper tape version plus the User's Manual.

For further information contact Tychon, Inc., Blacksburg, VA 24060. (703) 951-9030

8080 MICROCHESS

MICRO-WARE LIMITED is pleased to announce the availability of MICROCHESS for the 8080. MICROCHESS is an intelligent chess playing program which will run on any 8080 or Z-80 system with 4K of contiguous memory and an ASCII input/output device.

The MICROCHESS documentation includes a Player's Manual with a descrip- of the program, and a comprehensive appendix with details for conversion of the I/O routines, or customization of the board display programs. The program is furnished on paper tape.

Features of MICROCHESS include automatic board display, multiple levels of playing skill, castling, and the ability to reverse the board at any time, making it possible for the computer to play a game against itself.

MICROCHESS will be available at a price of $15.00 from most major computer stores or direct from:

MICROCHESS (8080),
MICRO-WARE LIMITED,
27 Firstbrooke Rd.,
Toronto, Ont.,
Canada, M4E 2L2.

DISK BASIC ETC

Binary Systems Corporation announces DISK BASIC ETC, a disk-accessing, extended version of BASIC ETC. DISK BASIC ETC, an interpreter for 8080-based microcomputers, was co-developed by John Arnold and Dick Whipple of Tyler, Texas, authors of the original Tiny BASIC. Arnold and Whipple also co-developed BASIC ETC, the forerunner of DISK BASIC.

Extensions to BASIC ETC include new options to extant commands; for example, with DISK BASIC ETC either a line number or executable statement may follow THEN or ELSE; the addition of ON-GOTO and EDIT statements; the inclusion of 11 more functions; and, refinements such as increasing the execution speed of some statements.

DISK BASIC ETC uses the lower 12 KB of memory plus 1 KB of scratchpad. User input and output routines, and stack and memory end values are specified in a user's manual.

DISK BASIC ETC is supplied on a certified, 5¼ inch minifloppy disk, or on a certified, 8 inch regular floppy, along with a comprehensive user's manual. The price is $50.00; the manual sells for $10.00 separately. Order from the Micro Store, 634 S. Central Expressway, Richardson, Texas 75080. Orders should include a check or money order for the price of the item.

TEXT EDITING SYSTEM

TSC (Technical Systems Consultants) is pleased to announce the availability of the first of several large-scale programs for the 6800 microprocessor. The new TSC Text Editing System is the most extensive text editor available to the micro user. It supports many standard commands, such as: PRINT, INSERT, DELETE, FIND, REPLACE, and VERIFY. These alone make up a respectable editor, but the following make it fantastic. Convenient pointer movers are provided for file TOP and BOTTOM, an APPEND command allowing any string to be appended to any or all lines starting in a specified column, text block COPY and MOVE are easily performed with a single command, and a very extensive CHANGE command allowing one to change any or all specified occurrences of one string into a second string.

Other features include: tab character definition, TAB column set, special character SET command, line NUMBERS on or off command, an EXPAND tab character command, easy pointer positioning using the NEXT command, a RENUMBER command, auto line numbering, STOP and LOG commands, as well as a unique OVERLAY command, allowing the user to conveniently change a line by typing over an existing line.

This list is not complete, but shows the completeness of the TSC Text Editing System. The editor is intended for those with serious needs. As with all TSC software, a complete source listing, hex dump listing, sample output, and complete user's manual are all provided. The price for all this is only $23.50 and delivery is from stock. Order number SL68-24, from TSC, P.O. Box 2574, W. Lafayette, IN 47906.

6800 MIKADOS

Inpro Micro Systems is pleased to announce the availability of a unique, low-priced Mini Instant Keyboard Assembler, Debug, and Operating System (MIKADOS) for developing small to moderate size programs on 6800 based microcomputer systems.

MIKADOS is specifically designed to minimize the amount of time and effort required to assemble, debug, and modify programs using a minimal amount of memory. It occupies only 2.5K bytes of memory. With only 4K bytes, this leaves 1.5K bytes which can be allocated for user programs and label table.

The assembler generates object code for the 72 basic variable-length instructions of the 6800 with all addressing mode variations. The assembler instantly generates object code for user-entered mnemonics. Formatted object code and address are printed on the same line as user input .... providing an immediate program listing. The object code is also loaded directly into user program memory. Relative branching instructions with symbolic labels are resolved by maintaining a label table for label prefixes and unresolved label operands.

All of this for only $12.95!!! (Price includes user manual and hex object code listing). Contact Inpro Micro Systems, P. O. Box 7776, Van Nuys, CA 91409.

PUBLICATIONS

$500 CHALLENGE TO CALCULUS BUFFS

A $500 cash award is offered to calculus buffs, science or math teachers, or other interested parties. The award will be paid by the publisher of NON-NEWTONIAN CALCULUS under the following conditions:

The publisher of NON-NEWTONIAN CALCULUS believes that the systems of calculus described therein were not known and recognized before 1967, the year in which the authors made their fundamental discovery. It is possible, however, that the authors were not the first to discover those systems. Accordingly, the publisher will award a *$500 cash prize* to the first person who submits suitable evidence that before 1967 those systems were known and recognized to provide inversely-related derivatives and integrals which yield numerical results differing from those of the classical calculus developed by Newton and Leibniz 300 years ago.

Furthermore, individuals may receive a full cash refund for the purchase price ($12) from the Non-Newtonian Calculus Institute, if the book is purchased through the distributor, Intergalactic Publishing Company, 108 Stratford Avenue, Westmont, N.J. 08108, Catalog No. NNC12.

10th ANNUAL COMPUTER BIBLIOGRAPHY

More than 225 new books are listed in the tenth edition of the *Annual Bibliography of Computer Oriented Books,* recently published by the University of Colorado. Notable in this year's listing is the increase in quantity *and* quality of books on the subjects of system design and program design. Also, the books on data communication increased by 50%. All books prior to 1971 were deleted, except for a few classics, despite the deletions, the bibliography still contains more than 1,000 books from 210 publishers.

The bibliography separates books into 55 categories and catalogs them according to type (reference, textbook, handbook) and style of presentation (P.I., case and normal). A new category on program design was added this year because of the increase of books on this subject; 10 new books were published in this category.

Continuing the trend of the past two years, the additions to the advanced programming area evidenced considerable improvement; the advanced programming category now contains more than 80 books.

A noticeable difference occurred in the design of many of the new programming books--they are oriented toward structured programming. It is now possible to find a S.P. book in every major programming language: PL/1, COBOL, FORTRAN, ALGOL and APL. Previously, structured programming was confined to books on PL/1.

Copies of the bibliography are available for $4 from *Computing Newsletter,* Box 7345, Colorado Springs, CO 80933. The cost is $5 if an invoice is required.

A SUCCESS OUT OF FAILURE

Ever wonder why nobody writes about computing projects which failed? Why every conference, every proceedings, every paper tells you how great it is (or is going to be), but nobody confesses that they've screwed up?

Well, Seattle computing devotee Bob Glass decided that confession might be good for the soul. So he wrote a book about failure, titled it *The Universal Elixer and Other Computing Projects Which Failed,* and it's just been published by Computerworld.

Projects Which Failed is a sometimes funny, sometimes sad series of stories about computing booboos. All of them are based on truth, with fictionalized names and places to protect the innocent (or, for that matter, the guilty!)

The book is available in paperback from Computerworld, 797 Washington St., Newton, Mass., 02160. The cost is $7.50, plus $1.50 postage and handling.

FREE BOOK ON MICROS

A new *Microcomputer Recipe Book* covers everything from soup to nuts, to put together your own microcomputer operating system for personal, business, or scientific use. Under *Ingredients,* there are a wide-variety of system components such as computers, semiconductor and floppy disk memories, CRT displays, and hard copy printers. Suggested *menus* for complete systems range from the "BIG MAC", a simple 8080-based computer hooked into the family TV set, to the "BEEF WELLINGTON" which includes a Processor Tech SOL 20 microcomputer, a dual floppy disk memory, and a DECwriter II printer as well as a video monitor. Whatever your gourmet taste demands, you will find it here. . . Bon Appetit!

In addition to the extensive menu of computer systems, the same recipe book includes a large selection of technical books relating to computer hardware design, software development, and "How to do it" books.

J. Benbow Bullock, Vice President
Computer Center, Inc.
321 Pacific Avenue
San Francisco, CA 94111
(415) 421-8686

MISCELLANEOUS

COMPUTER CAMP

Four one-week programs in computer programming will be offered this summer at Rose-Hulman Institute of Technology, Terre Haute, Indiana. The program, known as Camp Retupmoc, is boys about to enter their junior or senior years in high school; it consists of lectures on BASIC programming, films on computing, and talks by computer scientists in business and industry who are making novel applications of the computer.

Dates for the Camps are June 19-24, June 26-July 1, July 10-15, July 17-22. The fee, including tuition, room and board, is $125.

For further information contact Dr. John Kinney, Rose-Hulman Institute of Technology, 5500 Wabash Ave., Terre Haute, Indiana, 47803.

KIM SOFTWARE CONTEST!

1st Prize - KIM-3 8K Memory Expansion Board (value $289)

2nd Prize - KIMROM-1 Resident Assembler/Editor ROM set (value $150)

3rd Prize - 10th Prize KIMath Source listing and User Manual (value $15)

The KIM programming contest is open to all KIM owners and users. All entries must contain program documentation and source code listing (hand assembled source is OK). All entries become the property of MOS Technology, Inc. and will be turned over to the KIM Users Group for possible publication. Additional prizes may be awarded for noteworthy programs by beginners. Entries will be judged on the basis of originality and usefulness to the user community. If external hardware is required, a schematic should be provided. Complex programs taking more than 1K of memory, such as high-level languages, assemblers, cross-assemblers, text editors, etc., will be awarded a duplicate first prize if accompanied by working source tape or casette.

All entries must be received by July 1, 1977. Prizes will be awarded August 1, 1977.

Entries should be sent to:

KIM Software Contest
MOS Technology
950 Rittenhouse Rd.
Norristown, PA 19401

CALCULATOR FEATURES 1,500-HOUR BATTERY

Button cell batteries that provide 1,500 hours service time, Liquid Crystal Display and vest pocket size are three highlights of the newest National Semiconductor hand-held electronic calculator. The NS/100, which will carry a suggested retail price of $29.95, is about the size of a bridge playing card in width and length and measures only seven millimeters in depth. A full-function calculator, with accumulating memory, the NS/100 will also offer square root, "live" percent key and change sign.

Dick Veatch
Consumer Products Division
National Semiconductor Corp.
1177 Kern Avenue
Sunnyvale, California 94086
Tel. 408/733-2600, ext. 352

HAMFEST/COMPUTERFEST

The Jefferson Amateur Radio Club and the Crescent City Computer Club announces the New Orleans Hamfest/Computerfest will be held at the Hilton Inn in Kenner, LA (directly across from the New Orleans International Airport) September 24 & 25. This is the ARRL Delta Division Convention for 1977 and is the largest "Ham" outing in the deep south. This will not only be the largest Computerfest, but it is the *only* Computerfest in the area.

This year's event will feature a banquet Saturday night with entertainment, *two days* of commercial exhibits, fleamarkets and forums. There will also be a hospitality room, ladies' events, FCC examinations and more.

This year's grand prize is a complete Drake "C-Line" ham station and many door prizes will be awarded each day.

Information on tickets, room reservations and etc. will be furnished upon request by contacting the New Orleans Hamfest/Computerfest; P.O. Box 10111; Jefferson, LA 70181.

LETTERS

Letters to your editor have revealed some fairly prevalent confusions and/or misconceptions. Perhaps some historical notes will help clarify the situation.

Once upon a time ye gran ol' Dragon Bob Albrecht started a newspaper called *People's Computer Company* to help teachers and others learn more about computers. Later, a non-profit educational corporation of the same name was formed to publish the newspaper. Said corporation now publishes three computer-related periodicals. *People's Computers* is aimed at beginners and intermediates *Dr. Dobbs Journal of Computer Calisthenics and Orthodontia* is for heavy hackers — if you don't know what *that* means, it's probably not for you. *Computer Music Journal* is a state-of-the-art publication for specialists or would-be specialists in computer music. PCC (the company), also maintains a bookstore and occasionally publishes books such as *What To Do After You Hit Return: PCC's First Book of Computer Games* and *PCC's Reference Book of Personal and Home Computing*, Spring, 1977.

After 5 years of editing PCC's newspaper, the Dragon decided he was over-ripe, and so retired to Emeritus status. These days he's often found in his lair gloating over his treasures, breathing smoke and fire (both in and out), writing, and teaching kids. He's shared his treasures with us in recent issues; in this issue, see *Tiny Basic* and *Make Believe Computers*.

Since last October, I've been your editor. This is the fourth issue I've put together, the first with a new name and format. Changes you've seen over the past issues have been partly personal bias (I've designed and written tutorial CAI courses for more than a decade) but mostly in response to the results of a readership survey last fall. Keep sending in feedback: we really *do* care.

Phyllis Cole, Editor
People's Computers

Congratulations on your new format and new name! I hope you will go far in your new clothes. Thank you for publishing my letter, under the old PCC banner, in which I asked people to get in touch with me as part of my survey of humane computing. I received a number of interesting responses and my survey has been broadened as a result. While people generously shared with me their thoughts on humane computing, I was a little disappointed that I didn't hear from anyone who was actually *using* a computer in ways that they regard as humane. I think there are at least two reasons for this. One reason is that there simply are not many applications of computers that are humane, compared with the majority of uses that are neutral or negative in this regard. More importantly, however, I think it is that people are not familiar with the term "humane computing" and are unsure of how it can apply to what they do. Some may even regard it as inherently contradictory - which I trust it is not!

I would still like to hear from people in connection with my survey and I hope that by offering a simpler definition of "humane computing" and by giving some examples of promising applications, more of your readers will be encouraged to write to me. In general, what I mean by "humane computing" is the use of computers that directly enhances people's lives and that tends to build a more humane society. Whenever a person uses a computer out of their own free choice for what ever they want, then this is humane computing, at least at the personal level. The other part is to consider the broader social effects and see whether other people, now and in the future, will benefit. Hopefully the use of computers will tend to promote humane social and economic patterns. Within this broad definition, I am particularly interested in hearing from people who are working with small-scale computers that involve and serve more than just computer people.

Applications that are potentially humane could include:

- systems for creative expression e.g. in visual and musical arts
- small-scale business, accounting, mailing list, text handling systems
- information/communications networking. e.g. Community Memory
- small scale design aids
- small scale process and machine control
- handicapped aids, prosthetic devices
- games that promote exploration, understanding, and social interaction

especially when created for and used by people such as artists, farmers, neighbors, children, handicapped persons, students, musicians, small local businesses, cooperative enterprises, social active groups, and so on.

I would like to hear from people who have thoughts or experiences in this area that they would like to share with others. Any information that I receive will go towards a published report of my survey. I hope, in this way, that those of us who are concerned about the humane use of computers will benefit from each others experiences and help us in the development of tools that are of real service to people. Thank you.

Andrew Clement
789 West 18th Avenue
Vancouver, B.C.
Canada V5Z IWI

Do you have, or know of anyone that has, a BASIC system for the RCA 1802 micro-processor. I have a homemade job that could use BASIC. If you have such a system, could you inform me of the costs and memory requirements? I would prefer to get a listing of the program. I don't have a paper tape reader and my tape cassette is non-standard.

R. Alan Parker
9660 D Barrel House Road
Laurel, MD 20810

We have admired your projects from afar for some time. Now we'd like to help out with a retaining subscription. We are especially interested in the microprocessor language related articles.

Our primary business is microcomputer products for the small business area. We are very much in favor of distributing computer intelligence wherever possible and try hard to build useful systems that people will enjoy interacting with. We have a word-processing system on the way, with more business applications to follow shortly.

We try to spin off technological tools that we develop for ourselves to the hobbyist market at an affordable price. One example of this is our BLOS and ZAPS operating systems for the Digital Group 8080 and Z-80 systems respectively. We believe that these are the best integrated and most comprehensive non-disk operating systems available to the hobbyist. A description is enclosed. We have experienced demand to move these systems to other mainframes, and we will probably do this shortly. (Brent Longtin is primarily responsible for our operating systems development).

Our products are distributed through computer stores; currently we have seventeen stores in eight states. Other dealerships are welcome. I'll cut off there about our product line lest this sound like a commercial.

We always enjoy hearing from hobbyists and other people in this field. Our phone number is (617) 965-0545.

Bruce Cichowlas
Algorithmics Inc.
Box 56
Newton Upper Falls, MA 02164

Family Homes for the Adult Retarded, Inc., is a non-profit organization working with developmentally disabled adults. As is the case with most non-profit organizations, money – or more precisely, the lack of money – is a serious problem. We feel that one way to help alleviate this shortage, is to establish several small business concerns, that will generate some jobs, and some revenue for us.

We are interested in forming a computer co-op, and are seeking interested computer people. The co-op would work to establish computer games on a profit basis. We have some capital, but are interested in people with ideas, enthusiasm, and some money. All profits of the co-op would be divided among the members, with part being reinvested in the co-op.

Many people are interested in an adventure of this nature, but few of us could manage it alone. Together we might be able to succeed. We are not set on any particular type of computer operation, and are open to ideas. Interested people are urged to call me at (415) 593-2516 (office) or (415) 881-4759 (home).

Steve Berlin
Admin. Assistant
FHAR
901 North Road
Belmont, CA 94002

We just founded a Computer-Club recently in Zurich. At present, there are 40 members ranging from pure hobbyists to professionals.

The main goal of this Club is to establish a good communication between people which are interested in Personal Computing. We, therefore, would like to contact other Computer-Clubs to exchange information and experience.

For your information, the total microcomputer population here in Switzerland is estimated to be over 200 in private use for personal computing. These systems range from simple one card computers to full blown systems with floppy-disks and cassette tapes.

Applications are mostly games and small business processing. Some of the systems are used for research and development of new machines.

If anyone is interested to talk to us, just drop us a line. We will reply to every letter within 3 weeks.

Richard M. Vogel, President
Computer Club Zurich
Badenerstrasse 281
8003 Zurich
Switzerland

Results of the readership survey were most interesting. I found I was quite in agreement with the results of the "Reactions to PCC content". Looks to me as though many other readers have almost exactly the same interests I have. If *MY* interpretation is valid, it looks to me as though DDJ is *right on!* And, PCC *has been* rather wide of the mark. On the other hand, it looks to me as though the current issue represents a *shift in content.* I hope my perception is accurate and that the shift is not transitory because I LIKE WHAT I SEE. I spent more time reading this issue than I have ever spent with any previous issue of PCC. Keep up the good work!
James Lang
5 Beech Lane
Edison NJ 08817

The Delphian Foundation is a non-profit coeducational institution which operates the Delphian School in Sheridan. The school has been very successful in its approach to education: students average 19 months ahead of national norms in basic educational skills.
To respond to growing interest among students, the school recently expanded the curriculum to include computing technology. All courses demand from the student extensive application of the course materials. I should mention that the courses are defined by the ability the student gains, rather than by volume of the subject.
Presently I am working for the computer courses. I am collecting any materials I can get a hold of to evaluate it for usability on the courses. Material referenced in the courses will be studied in the original, so no copying or extracts of source material is required. I am particularly interested in exchange of ideas and experiences with individuals, who have a solid background in micro computers. My own background is not far off: I held all kinds of positions from service engineer to systems programmer and systems designer. I designed the data com network for a European airline with 1800 CRT's and some 400 TTY's, but have no experience with micros. I will have to learn about them!
Helmut Karl
Route 2, Box 195
Sheridan, OR 97378