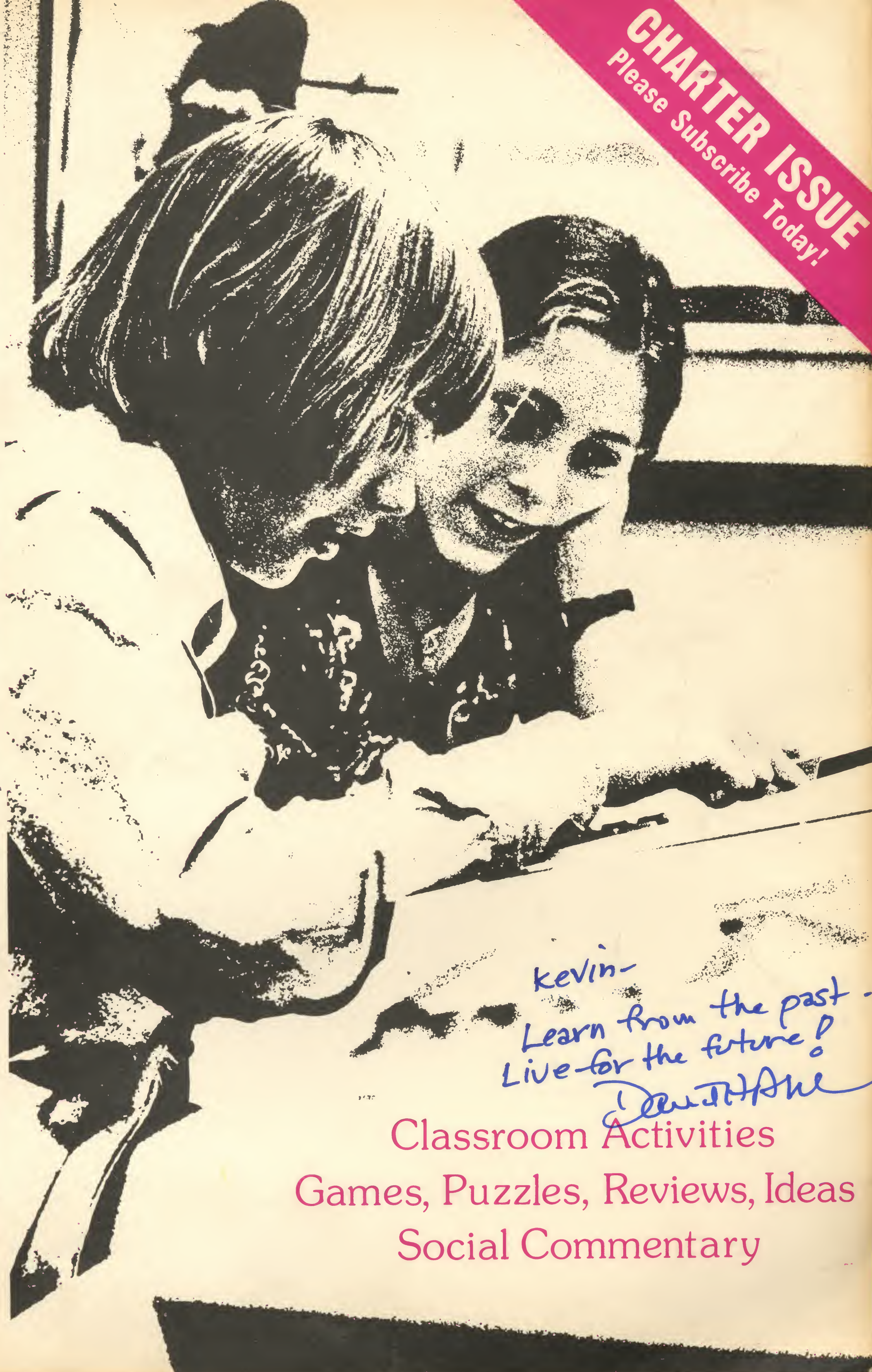
creative computing

a non-profit magazine of educational and recreational computing

Nov-Dec 1974

$1.50

CHARTER ISSUE
Please Subscribe Today!

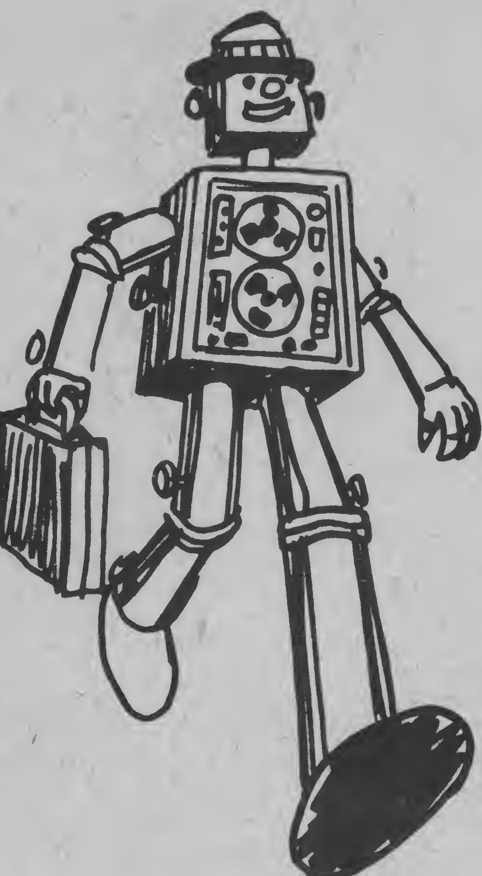


Kevin-
Learn from the past.
Live for the future!
David H Ahl

Classroom Activities
Games, Puzzles, Reviews, Ideas
Social Commentary

creative computing

★ a non-profit magazine of educational and recreational computing ★



Editor-in-chief and Publisher
David H. Ahl

Higher Education Editor
A. Kent Morton, Dartmouth College

Two-Year College Editor
John Tontsch, LA Pierce Jr. College

Problems Editor
Walter Koetke, Lexington High School

Reviews Editor
Lynn Yarbrough, Lexington, MA

Contributing Editors
Carole Cushing, Concord, MA
Bob McElwain, Sylmar, CA
Gwyn Lyon, Acton, MA
W. David Malcolm, Jr., Acton, MA

Contributing Artists
Carol Rasmussen, Concord, MA
Jeanne S. Roberts, Lincoln, MA
Kimberly Collins, Univ. of NH

Business Staff
Andree Stone, Concord, MA
Jane Weimann, Los Angeles, CA

Student Assistants
Dana Noftle, Acton, MA
Don Watson, Acton, MA

Volume 1 Number 1

CREATIVE COMPUTING is published bi-monthly by Ideametrics, P. O. Box 789-M, Morristown, NJ 07960. Third class postage paid at Morristown, NJ. Subscriptions: Institutional 1-year $15, 3-year $40; Individual 1-year $8, 3-year $21; Student 1-year $6, 3-year $16. Foreign postage $1.50 per year additional. Single copies $1.50.

In This Issue of CREATIVE COMPUTING

For the Computer Man illustration, thanks to Steve Rogowski, author of *Problems for Computer Solution* published by EduComp, W. Hartford, CT.

Assorted facts throughout this issue are taken from various United States Government publications, dictionaries, and "Incredible Facts, Amazing Statistics, Monumental Trivia" by Will Eisner, Poorhouse Press, New York.

THE COVER

The cover of this issue of CREATIVE COMPUTING is by Lisa Sheble, a teacher of art and photography at the Spence School, New York, New York. Kary Heuston is using the terminal while Janet Chaplan looks on. Both girls are in the seventh grade at Spence.

CREATIVE COMPUTING Editorial

A Computer in the Classroom?

A very large part of the General Public is probably not aware of the need for a computer in the classroom. The "successful" adult says, "What's all the fuss about? I got educated the old way. I learned a lot. I had some tough teachers and they made me toe the mark. I must say I've done well in life. Why do today's children have to be taught differently?" The "unsuccessful" adult says, "I was never any good in school — I just wasn't a brain. I was too dumb to get the stuff the way all the smart kids did. I never knew what it was all about. But that doesn't mean they should teach school for kids like me. There probably aren't many like me." Neither believes that a change is needed because neither believes what the more perceptive teachers know: that a very large proportion of math and science students "don't really know what it's all about."

Much that is reported in CREATIVE COMPUTING will be familiar to many members of the school science community. It should be. To find otherwise would be to find that they were insensitive to their surroundings. Why does it need to be said again? The ambience is not static. The atmosphere in which it is said today is not the atmosphere in which John Dewey spoke, in which Sputnik took flight, in which PSSC entered the schools. We urge a renewed awareness of familiar needs in today's ambience and an awareness of new needs arising from the pressures of today.

It is hoped that CREATIVE COMPUTING will focus attention on these needs, sharpen our awareness of them, and result in action.

Dave Ahl

Positions Available

CREATIVE COMPUTING has five editorship positions looking for someone to fill them. Each editor must prepare a column or article for every, or nearly every issue. The editor may write an original contribution or solicit it from an outside source. Positions are voluntary. They are open to teachers, students and others.

Careers Editor. Prepares a column focusing on careers in the computers field.

Interviews Editor. Secures interviews with leading educators, government officials, writers, and industry people.

Technology Editor. Prepares a column about the latest advances in computer technology, microprocessors, software, etc.

Elementary School Editor. Prepares one or more activities suitable for elementary age students.

Computer Games Editor. Checks out and runs computer games submitted to CREATIVE COMPUTING. Must have access to BASIC with very good quality printing terminal.

What's Happening?

In CREATIVE COMPUTING, we wish to offer you an opportunity to tell others about the things you're doing, the things you're interested in and anything that's on your mind. In our next issue, this spot will be filled with a feature article, written by a student. Your article can be printed here.

In "Speak Your Piece!", you can tell others what's on your mind. "Timewaster" and "Puzzles and Problems for Fun" both give you an opportunity to share your favorite nonsense. We invite you to send us the phrase for a "great" poster or a "terrific" cartoon picture. "Let's Take a Look at . . ." offers you the opportunity to share an interesting view of a topic others have considered. "Here's the Problem . . . " gives everyone a shot at it. In a contest, who knows what it takes to win.

The principle purpose of this part of Creative Computing is to provide students with opportunity to express themselves, to share their good ideas, whatever they may be. We hope you'll feel free to send us your material so that others may benefit from it.

Input/Output

Dear Editor:
Would you please send me some information about the games you can play on computers. If you would I would appreciate it alot. Thank you.

Bobbie Weaver
Fridley, MN

Dear Editor:
I am convinced a person can learn more by doing game programs requiring imagination than just writing out cut-and-dried problem-solving programs. The former, to put it crudely, is just more fun than the latter. And I get the impression that CREATIVE COMPUTING realizes this too, so I'd better subscribe.

Ricky James Roberson
Cleveland, TN

Dear Editor:
I want you to know that my office stands ready to be of all possible assistance should CREATIVE COMPUTING request U. S. Government assistance.
I send my hopes for every success in your efforts.

Edward M. Kennedy
United States Senate

Dear Editor:
I read about you guys in Computer Decisions and would appreciate any info. This sounds like a very worthwhile concept.

Douglas W. Champion
North Carolina State

Dear Editor:
Please send subscription information. I am a data processing professional and welcome the chance to inform my children of my profession.

Edward P. Evans
Sacramento, Calif.

Dear Editor:
The folks you have gotten together in this read like a who's who! Good luck to you all in this venture.

Dietrich VonSchwerdtner
The Peddie School
Hightstown, NJ

Dear Editor:
We agree that a more scholarly journal for the uses of computer-oriented instructional systems is needed.

Barry L. Bartlett
TICCIT Project
Brigham Young University

Dear Editor:
I would be glad, speaking for SIGCUE, to cooperate with you in any way we could to assist you getting it started.

Alfred M. Bork
Chairman, ACM SIGCUE
University of California, Irvine

Dear Editor:
CREATIVE COMPUTING – good idea! Happy to contribute articles.

John M. Nevison
Project CONDUIT
Dartmouth College

Dear Editor:
I think CREATIVE COMPUTING is a good idea . . good luck, and thanks for the chance to comment.

Bill Mayhew
Computer Systems Director
Boston Children's Museum

Dear Editor:
Your idea looks interesting. Good luck in bringing it to a successful birth.

Fred W. Weingarten
Division of Computer Research
National Science Foundation

Dear Editor:
Basically the idea is good and I think such a journal would be very useful for the dissemination of computer-based materials. I see a regular publication as a logical – and convenient – means to provide for transportability of programs and pedagogical strategies.

J. J. Lagowski
Director, Project CBE
University of Texas, Austin

SEXISM IN COMPUTERS

A computer in the USSR did all right by male mathematicians but began giving trouble when a woman tried operating it. Both male chauvinist pigs and women will be disappointed to learn that the reason was not that the lady's input was unacceptable to the computer because of its alleged feminine illogic or that the computer became overheated at the sight of her beauty, but rather that her dress was made of synthetic fiber producing an electric field that affected the computer.

BRAIN TEASER

Have children go to the chalkboard for this activity. Then give them the following instructions: "Write down your age (say 11); multiply by 2 (22); subtract 3 (19); multiply by 50 (950); add 39 (389); add the change – under $1.00 – in your pocket (989+32=1021); and add 111 (1132)." The children will be surprised to find that the first two digits in their answer always equal their age and the last two digits always equal the amount of change in their pockets.

Why?

The Guiness Book of Computer Records

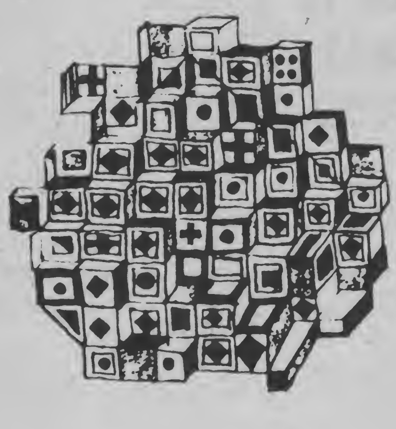
by Lynn Yarbrough

It would be interesting to add a chapter to the Guiness Book of World Records on the topic of computer programs. What is the biggest computer program ever written? The longest running? The most complex? The most famous? The most used? In order to establish records for these and other categories, we must first agree on what we mean by some of these. For example, we probably can agree on a criterion for complexity: the total number of conditional jumps is an obvious one. But to eliminate programs constructed solely for the purpose of breaking records, we would also have to establish some criterion for usefulness — and that may be difficult to get people to agree on.

Here are some candidates for records, based mostly on my own limited knowledge and personal criteria for usefulness. I am eliminating operating systems from consideration, because they are intended to run forever, or at least until the computer conks out, and concentrating on programs that are designed to solve specific problems.

Biggest program —
The Project Apollo real-time system. I don't know quite how big it is, but it's on the order of 1,000,000 lines of code.

Most complex —
Maybe Project Apollo again, but for density of conditions I suggest that PL/I compiler may have a higher percentage of conditional jumps. Any other candidates?



Most famous —
The COBOL compiler on the IBM S/360.

Most used —
The control programs for the Bell Telephone Companies' Electronic Switching System.

Most subroutines —
I had heard, a few years ago, of a FORTRAN program with just under 200 subroutines, but I can't claim a record for that because I can't identify it. Has anyone a candidate for this category?

Longest running —
Prof. Donald Knuth of Stanford University once ran a program to exhaustively analyze a chessboard problem. The program was run at odd hours when the computer was otherwise idle, and thus took over a month to complete. In total, the run used about 180 hours of central processor time — on a CDC 6600! But the Atomic Energy Commission people may have this record beaten already — can anyone fill us in?

If instead of only programming records, we extend our new chapter to include other computer-related records, here are some other items for consideration.

The largest punched card file in the world was for many years maintained by the U. S. Bureau of the Census. After the census of 1960 this file contained over 500,000,000 cards. Consider the amount of storage space for this amount of cards; packed as densely as possible this represents a stack of cards over 100 miles long! (I wonder if this file is still maintained, and if so how big it got after the 1970 census.)

The largest magnetic tape file in the world is maintained by NASA. It consists of all the telemetry and other data gathered from all the missions, both successful and otherwise, flown by U. S. rockets and satellites since the beginning of the space program.

(NASA has a major problem in using this data before the tapes disintegrate. There is so much data that 90% of it will probably never be examined by anyone.)

What facts can you add to the CREATIVE COMPUTING book of programming or computer records? Who can tell us more about the widest magnetic tape (3" tape used by an early Honeywell computer), or the amount of heat generated by Einac or Whirlwind, or the most computers tied together (ARPA Net?). As you give us information, we'll try to tie it together and publish the results from time to time.

Another new classroom activity from CREATIVE COMPUTING Magazine ......

INTERVIEW

Here's an informative activity to do in a "Computers in Society" or "Computer Appreciation" course, or for that matter, in a social studies or sociology course.

EXERCISE 1

Make up copies of the interview form on the next page and give each student two copies. (You may want to use a subset of the questions instead of the entire list.) Each student should fill out one copy of the questionnaire himself. Then, each student should interview an adult on these issues. Try to obtain interviews with a diverse cross-section of people. Students may feel more comfortable working in groups to get interviews; if so, let them pair off. But no more than two students to a group; more than that tends to overwhelm interviewees.

EXERCISE 2

Tabulate the results, compare the various answers obtained, and discuss in class. Can you draw any conclusions about the attitude of the general public toward computers? Do students' attitudes generally agree or disagree with those of the interviewees? Are there any obvious relationships between the attitudes expressed and the demographic characteristics (age, sex, etc.) of the respondents?

Send your raw results to CREATIVE COMPUTING, i.e., number of total responses in each box, and we'll tabulate and print them all in a future issue.

OPTIONAL EXERCISE 1

Write a computer program to tabulate the results and compute average scores for each question as well as percentage distributions.

OPTIONAL EXERCISE 2

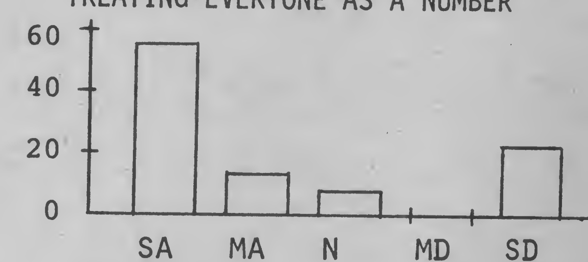
If your computer system has file capabilities, write a program to administer the questionnaire via a terminal, store the results and merge them with all previous results and then print out the scores to date. For real pizzazz, print the results graphically like this:

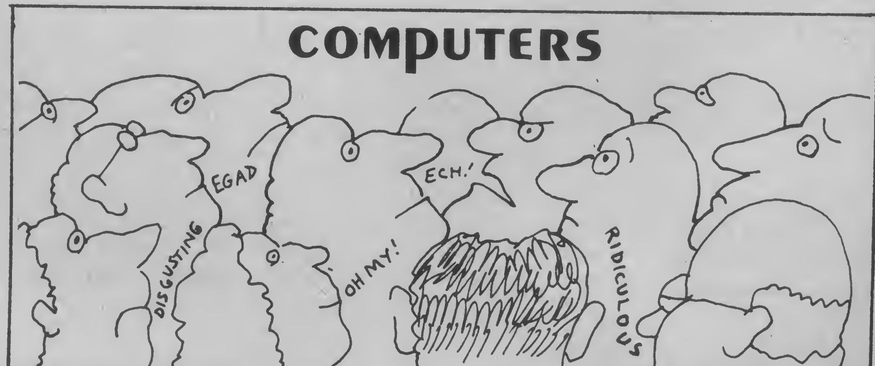
```
COMPUTERS WILL IMPROVE HEALTH CARE

SA      MA      N       MD      SD
*               A
|               └──── Average to date
└── Your answer
```

Or, show a bar chart of answers like this:

COMPUTERS DEHUMANIZE SOCIETY BY TREATING EVERYONE AS A NUMBER





CREATIVE COMPUTING
COMPUTERS AND SOCIETY QUESTIONNAIRE

	Statement	Strongly Agree (1)	Mostly Agree (2)	Neutral or No Opinion (3)	Mostly Disagree (4)	Strongly Disagree (5)
1.	Computers will improve health care.					
2.	Computers will improve education.					
3.	Computers will improve law enforcement.					
4.	Computers slow down and complicate simple business operations.					
5.	Computers are best suited for doing repetitive, monotonous tasks.					
6.	Computers make mistakes at least 10 percent of the time.					
7.	Programmers and operators make mistakes, but computers are, for the most part, error free.					
8.	Computers dehumanize society by treating everyone as a number.					
9.	It is possible to design computer systems which protect the privacy of data.					
10.	Credit rating data stored on computers have prevented billions of dollars of fraud. This is a worthwhile use of computers.					
11.	In the U.S. today, a person cannot escape the influence of computers.					
12.	Computers will create as many jobs as they eliminate.					
13.	Computers will replace low skill jobs and create jobs needing specialized training.					
14.	Computers are a tool just like a hammer or lathe.					
15.	Computers are beyond the understanding of the typical person.					
16.	Computer polls and predictions influence the outcome of elections.					
17.	Computers isolate people by preventing normal social interactions among people who use them.					

Age_____ Sex_____ Education_____________
Occupation__________ Location___________
Name (optional)_________________________

CREATIVE COMPUTING
P.O. Box 789-M
Morristown, New Jersey 07960

Computers

Statewide pools may not yield expected benefits

ST. LOUIS

The pooling of computer services in statewide networks can provide economies and efficiencies, but it often means loss of local control and may make computer services more expensive and less accessible for some users, concludes of a 12-month study in the U.S. and Canada by Charles Mossmann, director of user services at the University of California at Irvine. Mr. Mossmann conducted the study under a grant from the Exxon Education Foundation.

"The 'network bandwagon' is highly evident in the state and provincial capitals," he said, "and bureaucrats and administrators are scrambling to get on it.

"In nearly one-third of the 60 states and major provinces, decision-makers are guided by an explicit plan—generated at the provincial or state level—for computing in higher education. This plan determines what they will and will not do. In another third, such a plan is now under development.

Sharing of Resources

"Almost all such plans explicitly call for the sharing of resources between campuses, usually by means of a network constructed specifically for this purpose.

"Only 20 states and provinces indicate that their public colleges and universities are totally responsible for their own decisions about computing."

Unless there is a strong program of service and consultation for the "naïve and long-distance users," Mr. Mossmann said, state systems tend to be monopolized by the central, on-campus users.

Some network systems are successful, he said, mentioning Dartmouth, Iowa, and North Carolina as examples.

Cooperation a Goal

"A return to the 'one-campus, one-computer' model of the 1960's is just out of the question," he said. "The alternatives simply provide too many opportunities for both quality and economy of operations.

"I think it is not an unreasonable goal to strive for meaningful cooperation between colleges and universities, for networks that will spread resources to the have-nots and that will enrich the opportunities available to those in a position to use them.

"This cooperation can emerge between institutions and should not have to be imposed on unwilling colleges by an authoritarian governance.

"Computing, in its current stage of development, is uniquely a substance that can be shared. If we are the custodians of the first academic resource which can in fact be shared despite geography, and are the first agents to attempt such sharing on a large scale, it is no wonder we are having trouble. We have no model on which to build. In fact, we may be constructing models that have importance and implication beyond our vision."

Mr. Mossmann discussed his findings at a meeting of EDUCOM—the Interuniversity Communications Council of Princeton, N.J.—which sponsored his study.

Computers tend to be unpopular on campuses, said Ronald Roskens, chancellor of the University of Nebraska at Omaha, and he blamed higher education for allowing alienation and antipathy toward computers to develop among faculty members and students.

"We are suffering the consequences of a fairly high degree of computer illiteracy," he said.

"In my judgment," he continued, "there is no single force that has had greater impact upon the style and the operation of American colleges and universities in the last half-century than computer technology."

Although he found a lack of trust in computers within the academic community, Mr. Roskens assigned computers an important role in helping higher education to regain public trust through improvement of institutional management.

The Omaha chancellor suggested that computer personnel could improve their image by avoiding exaggerated claims of what computer systems can do and how fast they can do it and by avoiding "esoteric jargon."

Every college graduate ought to have some degree of computer literacy, said Gerard Weeg, director of the University of Iowa computer center and chairman of the EDUCOM conference.

Proliferation of Small Computers

He said there are approximately 10,000 computers in use in educational institutions now. It has been predicted that by 1980 computing may be the world's largest industry, he said.

It's hard for bureaucracy to keep abreast of technology.

A computer problem discussed at the conference here was reminiscent of the multiplication of copying machines that hampered attempts to centralize campus printing.

Now, directors of centralized computer services, see a threat in the proliferation of small machines of increasing sophistication.

Calculators have developed from mechanical adding machines to electronic, integrated-circuit packages that will perform all kinds of mathematical chores, some of them following computer-like programs fed to them on magnetic tape or cards, and costing up to $12,000.

Leland Williams, president of Triangle Universities Computation Center, said that one of the universities in his group now had about five programmable calculators that slipped through a screen designed to guard against the proliferation of computer facilities.

"When is a programmable calculator a computer?" Mr. Williams asked. —JACK MAGARRELL

Reprinted with permission from The Chronicle of Higher Education, Aug. 5, 1974.

Adding up calculators

The electronic calculator became a billion-dollar market at retail last year, according to a study by Creative Strategies. The biggest share of the market was held by consumer calculators, of which 7 million were sold. Some 3.5-million of last year's unit sales were business calculators, and another 300,000 were in the "professional" category. The study indicated that the market for consumer calculators would grow about 50% this year, in terms of dollar volume, more in terms of units, due to declining prices. Component costs for calculators have dropped sharply, LC chips falling from an average of $30 in 1970 to about $5 currently. LED displays, which cost calculator manufacturers slightly less than one dollar per digit, will be closer to 50 cents before the year is over.

EVELYN ROTH

Who is Evelyn Roth you may ask. Born in Mundare, Alberta, Evelyn grew up on a little "cow-wheat-chicken" farm and attended a one-room school until the age of fifteen. She then moved to Edmonton, Alberta, where she completed her education and began working in libraries, taking dance and fencing classes and designing clothes.

In 1961, she moved to Vancouver and became part of INTERMEDIA, a group of artists, musicians, film makers, and dancers who shared both common ideals and work space. In 1969, Evelyn worked with the University of British Columbia School of Architecture on a six-month project in Venice.

On her return to Vancouver, she began to channel all her artistic efforts into sculptural-wearables (multi-functional clothing) and dance-theater events. All of her sculpture-wearables are loosely knit from soft, flexible materials to provide both sensuous enjoyment and maximum flexibility to the wearer. Her ecological and environmental awareness and concern is reflected in her choice of materials. Working primarily with discarded fabrics, Evelyn Roth transforms throwaways into objects of fantasy and invention.

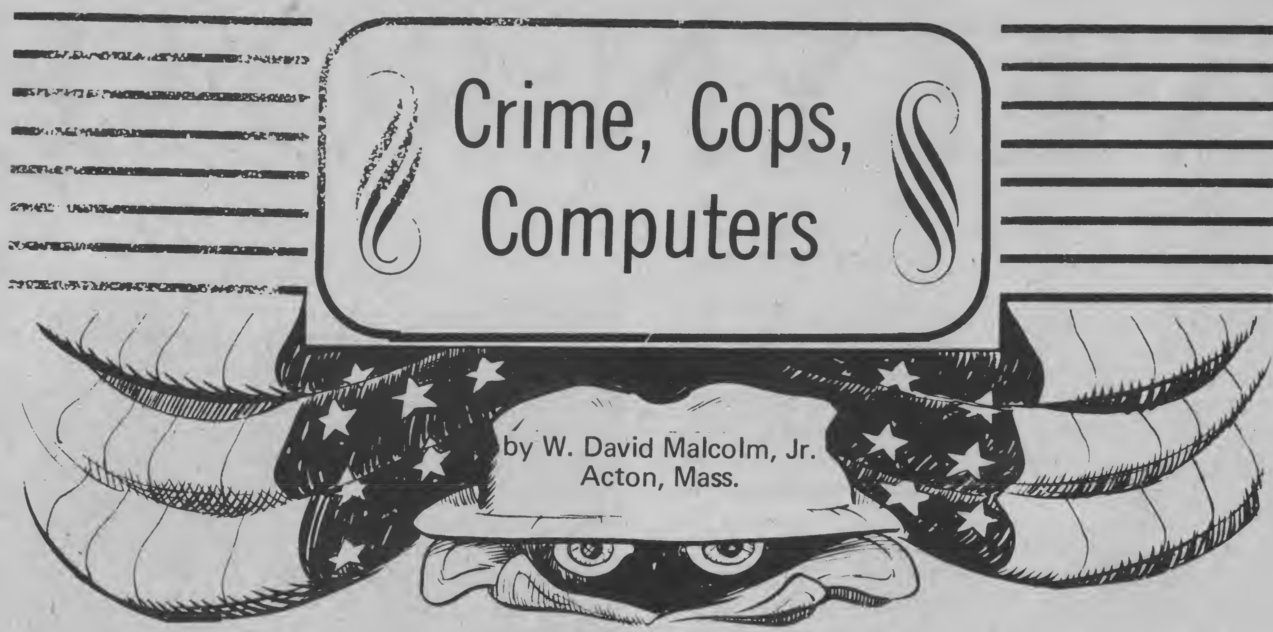
Evelyn Roth at the British Columbia pavilion at Expo '74 displaying her video tape creations.

Why, you may wonder, is CREATIVE COMPUTING writing about Evelyn Roth? Because Evelyn has found a new, unique use for both video tape and computer tape. Video tape can only be used 300 to 500 times before it loses reproductive quality and must be discarded by TV studios. Evelyn collects this tape from TV stations in the U. S. and Canada and crochets them into many articles like hats, bags, slippers, costumes, car cozies, and canopies. During the month of June 1974, she, with the assistance of Marion Denny, crocheted a giant canopy (1500 sq. ft.) using over 1 mile of tape to provide a shaded area in the courtyard adjoining the British Columbia pavilion at EXPO '74.

Evelyn emphasizes that "the material is fireproof and checked by many fire marshalls. That's a good point, because most people think it burns like film."

Why not try crocheting a sculpture-wearable yourself out of those reels of computer tape that are starting to drop more bits per inch than they retain? Send us a photo of your creation, and we will print it in a future issue of CREATIVE COMPUTING.

Have you ever seen a better use for discarded computer tape than a sun shade canopy?



Crime, Cops, Computers

by W. David Malcolm, Jr.
Acton, Mass.

A car had been burglarized. The detectives didn't have much to go on — only that three men had been seen leaving the scene in an "old tan-and-white station wagon." No make, model, year, or license number. It was clearly a case for PATRIC, the new detective's helper at the Los Angeles police department.

PATRIC (for "Pattern Recognition and Information Correlations") is a computer system that does the same kinds of things that a detective does, but does them much faster. PATRIC is crammed full of criminal records, crime reports, information on stolen vehicles, even the favorite methods of known criminals. By instantly cross-checking bits of information fed into it, PATRIC can quickly build up more and more information, and eventually come up with likely human suspects.

In this particular case, PATRIC searched its files, and found another car crime in a different part of the city, also involving men fleeing in a tan-and-white vehicle — but this time someone had remembered part of the license number and reported it. Using this partial number, PATRIC found the names of five men who had been stopped for questioning in similar cars. The computer then searched another file on past arrests, and found that three of the five men had previously been arrested for theft from an auto! PATRIC turned over the names to the human detectives, who promptly investigated and then arrested the trio for the latest burglary.

PATRIC took 15 minutes to produce the suspects; a detective would probably have decided the case was not worth spending hours or days sifting through all that information, with the likelihood that suspects couldn't be found anyway. Even when there is more information available, the computer can save hours of detective work.

Until recently, the men in blue couldn't afford much new equipment. But now they're getting funds from the federal government's Law Enforcement Assistance Administration (LEAA), a branch of the Justice Department. LEAA is pumping about $800 million a year into the various states, much of it going to their police organizations for equipment like PATRIC. These systems are perhaps the most significant elements of technology aiding police today. One of their biggest contributions has been a vast improvement in communications, which has always been a major problem.

In a noncomputerized police department, the dispatcher at the station house is flooded with radio calls from the cruisers, crime reports coming in by telephone, and other messages. During peak crime hours the dispatchers are frequently overwhelmed; sometimes police radio channels get so clogged that the policemen can't get through to the dispatcher half the time.

Now many police forces are installing computer terminals in patrol cars. Connected by their own radio frequencies to central computers, the terminals allow patrolmen to bypass the overworked dispatcher when they want certain kinds of information, like license numbers on hot cars and rap sheets on suspects picked up in the field. A cop in the cruiser queries the computer using a keyboard on the terminal, and gets an answer in seconds. In one case, two patrol cars carrying test terminals for a month recently made seven times as many "hits" on hot cars as they did without them. This was because they were able to make many more inuqiries through the computer than any human dispatcher could handle.

Checking Everything in Sight

The sheriff's department in Palm Beach, Florida has been using in-car terminals for almost a year, and patrolmen hare having a field day catching car thieves. They just keep poking thousands of license numbers on everything in sight into their computer system to see if the vehicles have been reported stolen. When it's quiet, they go through motel and other parking lots looking for numbers. Since the terminals went in, stolen car recoveries have climbed 60%!

The Palm Beach system uses solid-state plasma display and keyboard terminals that are located in each patrol car, hooked up to a 28K, 1.2 million bit PDP-11 controller in the main communications center. The inquiry/response mode is activated by the patrolman pressing special keys which are coded for particular messages. Requested information on, for example, whether a car is "hot" or not, who is the registered owner, and is the person wanted, is received back on the display in six to eight seconds. This response time compares with voice transmission (such as "Harry, could you query the computer regarding a John Jackson?") turnaround time of 90 seconds.

Besides the obvious advantage in rapid apprehension of criminal suspects, this reduction of time delays is also saving officers' lives. When the information turns out to be a "hit" (car or person wanted), an alarm is rung from the communications center. The dispatcher then reports the location of that patrolman on the displays of all cars in the field; the investigating officers also knows to use caution in the apprehension.

Another safety device is the emergency key. When an officer stops a car, he first enters the time and location on the terminal. If, at the end of three minutes, he has not reported back to the control center via the terminal, an emergency signal sounds in the center, and the location of the officer, who presumably needs help, is flashed onto all other terminals in the field.

Depending on the circumstances, the officer in the field can get a great deal of information concerning a vehicle or a person under suspicion. One interesting case occurred when the computer was first installed in Palm Beach. Someone standing outside a closed gas station at 2 a.m. looked suspicious to the road patrolman. So the patrolman stopped and asked "What are you doing here?" and the man replied "Just waiting for my buddy, just going to go fishing." The patrolman got his identification and entered it in the terminal in the car, and it came back that the guy was not wanted. So the patrolman queried the terminal if the fellow had an arrest record. It came back that the guy was a specialist in breaking into gas stations. So the patrolman went back to the man and asked, "Have you ever been arrested?" The man replied, "No, I have never been in no trouble, I'm just going fishing." The the patrolman brought the guy over to the car and said, "Do you see that screen?" And the guy said, "Oh, s---." Needless to say, he had already been in the gas station and was on his way out when he got caught. He just copped out right there, and that was the end of the ball game!

References:

The Wall Street Journal, Monday, March 18, 1974 *"Modern Detection* – Police Weapons Range From Electronic Cops To Glowing Bacteria"

Datamation, September 1973, pp. 88-90 "Law Enforcement Communications Conference" by Wendy Reid.

IN THE 1960's THE U.S. ARMY EMPLOYED OVER 1,500 PLAINCLOTHESMEN WHO WERE REPORTING ON INDIVIDUALS TO ITS MANY DATA BANKS

Is Big Brother Watching You?

The NATIONAL CRIME INFORMATION CENTER (NCIC) may have information about you. To find out what the FBI knows about you, send $5 and a set of rolled-ink fingerprints (which you can get from your local police station) taken on a fingerprint card, and containing your name and birth date, to: FBI, Identification Division, Washington D.C. 20537. Inaccurate data on your record can be corrected by contacting the agency that originally provided the information to the FBI.

Small School Makes Big Use of Mini

Introduction

Less than a quarter of a mile from the famous Old North Bridge in Concord, Massachusetts, where "the shot heard 'round the world" was fired, is the picturesque locale of the Fenn School. Fenn School for boys was established in 1929 and at the present time has an enrollment of 187 students in grades 4 through 8. This is the story of the first year with a minicomputer incorporated into the curriculum.

Past Experience With Computers

Periodic Visits:

A small number of the eighth grade students have had as much as two previous years experience with computers and programming, but the majority are learning as beginners. "We started with computers five years ago," explains Bill Maxwell, Head of the Mathematics Department. "At that time I took 10 to 12 of our most interested students to the Middlesex School, where they could acquire some experience using Middlesex's minicomputer. Computers are a fact of life and I feel the students gain valuable experience using them."

Teletype/Telephone Installed:

Initial experience with a computer excited the students to such a degree that the following year Fenn installed a teletype and telephone line directly to Middlesex. During off-hours, the computer was time-shared by the boys. This eliminated trips back and forth between schools and maintained the students' interest.

Timesharing Service Employed:

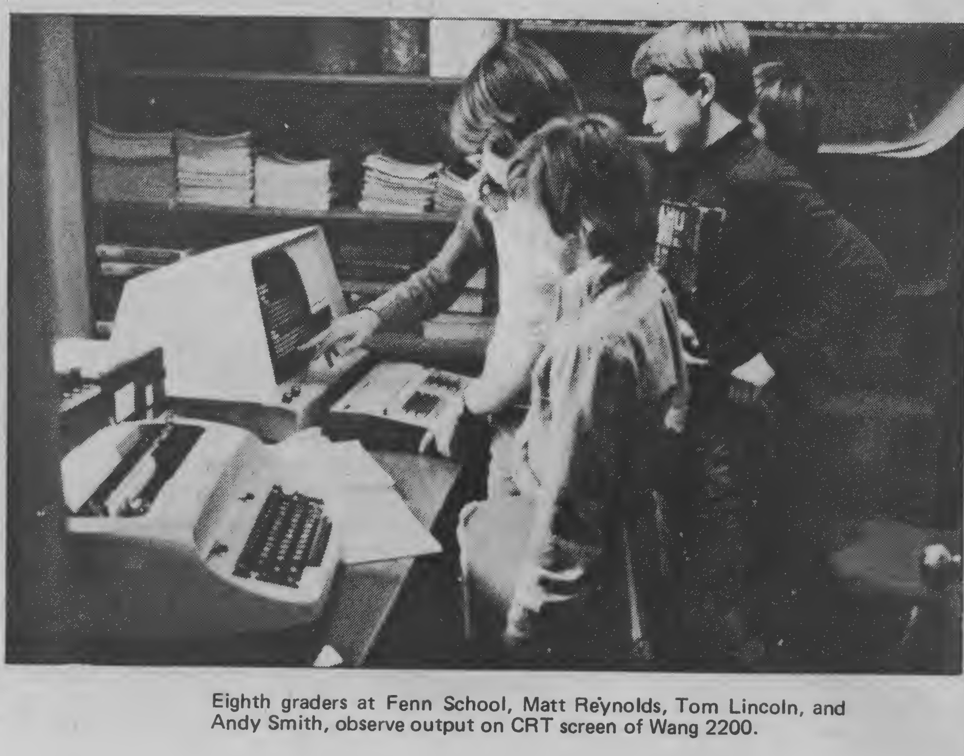
In order to provide further flexibility in time scheduling, Fenn subscribed to a local time sharing service for the next two years.

Unfortunately, the school was dissatisfied because of the lack of service supplied by the time sharing bureau. Machinery and telephone lines were constantly breaking down. "This initiated the desire to separate ourselves from several middlemen," stated Maxwell. "We wanted to cease dealing with the telephone company and with hardware problems 20 miles away. When students are held up waiting for repairs, frustrated in their attempts to try freshly-written programs, their enthusiasm wanes. Young students in particular must be rapidly reinforced in order to keep their interest high."

Acquisition of a Minicomputer

In the spring of 1973, Bill Maxwell at Fenn had an opportunity to evaluate the Wang 2200 and decided "that this machine was exactly the kind of system I wanted for these kids." "An on-site system has definitely excited our students; they're really turned on by it," says Mr. Maxwell. "They get in the room with the machine and they know exactly what they're going to do — and it works, that's the main thing, it works and it's reliable!"

One of the aims of the Mathematics Department is to have the boys know how to expertly use the system with the BASIC language by the time they graduate 8th grade. The best proof that a student has mastered a mathematical



Eighth graders at Fenn School, Matt Reynolds, Tom Lincoln, and Andy Smith, observe output on CRT screen of Wang 2200.

concept is if he can teach it to someone else. By programming the 2200, the student must "teach" the machine. When the system displays the student's answers on the CRT screen or types them out on the output typewriter, not only is his learning immediately reinforced, but proof of his mastery of a mathematical concept is demonstrated.

"The primary purpose for having a computer program is to use it to help the boys learn how to solve problems," explains Maxwell; "This is accomplished by having them (a) analyze the problem, (b) organize their thinking (flow-charting), (c) "play computer" with some examples, (d) finally run the program to verify its accuracy, and (e) submit a printout to the teacher to show that they have indeed done it."

Future Plans

All four math teachers in grades 7 and 8 will use the system throughout the year in their classes, integrating the computer with the regular math topics whenever they can, for reinforcement.

It is fairly certain that the 6th graders will also make use of the computer. The 4th and 5th graders may even get an opportunity to use the computer this year. If they do, they will begin in much the same way as do the older boys, i.e. by writing simple programs such as generating series, functions, guessing games and math quiz games.

Although the main reason for purchasing the computer system was for direct use in student education, some administrative applications are being explored.

It is presently being used to automatically type form letters, which saves the school secretarial or auto-type service costs. Other applications may be to incorporate alumnae records and class schedules on cassette tape. What next?

RESEARCH STUDY NEEDS HELP

The American Institutes for Research, under a grant from the National Science Foundation is conducting a nationwide survey on computer usage in secondary schools to provide a current assessment of the extent and nature of computer usage for instructional and administrative purposes. Questionnaires will be mailed to a random sample of high school principals in October. Since the survey is national in scope, all schools selected for the survey, computer users or not, are urged to respond.

Should your school not be selected as a participant, but you would still like to report how you are using computers in your high school, send a description of your activities to:

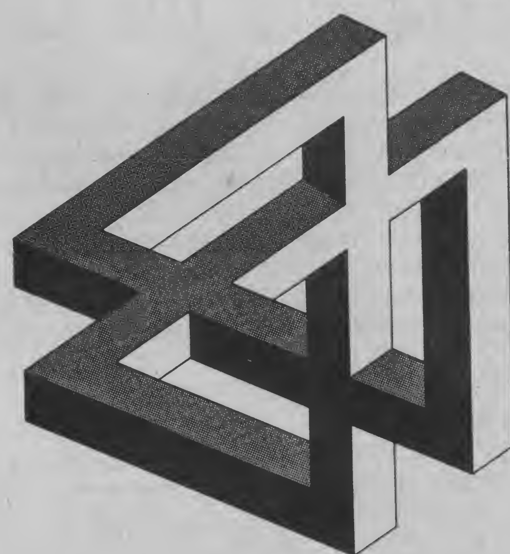
Dr. William J. Bukoski
The American Institutes for Research
Communications Research Group
3301 New Mexico Avenue, N. W.
Washington, DC 20016

Your participation in this study would be most appreciated.

Problems for Creative Programming

Assume a life-span of 80 years. In what year of the 20th century (1900-1999) would you have to be born in order to have the maximum number of prime birthdays occurring in prime years? The minimum number?

Marsha Lilly
Sudbury, Mass.



A regular dodecahedron has twelve pentagonal sides and twenty vertices. Assuming that one face is in the X–Y plane with an edge along (0,0,0) to (0,1,0), what are the coordinates of the remaining 18 vertices?

Lynn Yarbrough
Lexington, Mass.

There are 720 ways to arrange the digits 1 through 6 as six-digit numbers:

1 2 3 4 5 6
1 2 3 4 6 5
1 2 3 5 4 6
etc.

If you continue this sequence, in numerical order, what will be the 417th number in the series? What will be the *n*th?

Bill Morrison
Sudbury, Mass.

Problems for Creative Computing

by Water Koetke

The problems to be discussed in this column are those that seem particularly well suited not just for computing, but for creative computing. They will cover a wide variety of topics and subjects, and all are intended for both students and teachers — for anyone turned on by challenging problems, games or programs.

Your reactions will be very much appreciated. Suggestions for future columns, solutions to problems discussed, new problems, extensions and experiences with problems discussed are all solicited. Please address all correspondence to Walter Koetke in care of *Creative Computing.*

The challenge of creative thought is before all of us — this column is intended for those who choose to demonstrate that creative thought is also behind them. I hope you find the ideas rewarding.

Tac Tix and the Complications of Fallibility

The game of Tac Tix was created by Piet Hein, also the inventor of Hex, in the late forties. A first impression of the game is likely to be that it is indeed simple, but first impressions themselves are over-simplifications aren't they? The rules of Tac Tix are few, a desirable characteristic of games to be used in the classroom.

Each game begins with 25 markers arranged in a 5 x 5 square formation as in the diagram.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Two players then alternate turns. On each turn a player may take as many markers as he chooses from any single row or column, provided that the markers are next to each other. For example, markers 1 and 3 cannot both be removed on a single turn unless marker 2 is present and is also removed with the others. The player who removes the last marker is the winner.

STOP READING this article. Put it down and take a few minutes to analyze the game. The first player has an easily described winning strategy. Can you find it?

Assume that the first player can play without error. On his first turn he should remove marker 13, the center marker. On each subsequent turn he should remove the markers symmetrically opposite those removed by his opponent. By playing in this manner he is assured of winning the game. After the center marker is removed, a typical game might be:

Second Player	*First Player*
2 - 5	21 - 24
15, 20, 25	1, 6, 11
8 - 10	16 - 18
7	19
14	12 (wins)

When playing this game, an equally infallible second player is likely to be bored to death.

Since Tac Tix is played on a small board, has only a few easily stated rules, and requires only a short time to play, it is a very good game to implement on a computer. Writing a program that will play Tac Tix with a user by following a well defined strategy is an excellent problem at two different levels.

First, try a program in which the computer is the first player. To do this, one must be able to create a program that: represents the 5 x 5 board using single or double subscripted variables; makes symmetrical moves; determines if the second player is making a legal move; and realizes that the game is over. When writing the program one faces many of the likely difficulties encountered in far more complex problems.

Second, try a program in which the computer is the second player. If all users were infallible, then this really isn't worth writing. However, somewhere there may be a student or teacher who occasionally makes an error. Assume that you're writing the program for him. By considering this small bit of reality, a trivial case in the world of perfect people has become a rather challenging, interesting problem.

Consider each of the following opening plays of the first player. What is the best counter play for the second player?

Opening Play	*Counter Play*
3, 8, 13	?
16 - 18	?
11 - 15	?

In general, the second player should attempt to play so that for every missing marker the symmetrically opposite marker is also missing. The center marker must also be missing. If the second player succeeds in obtaining this board configuration at the end of any turn, he has successfully taken advantage of the first player's error and has a winning strategy. On all subsequent turns he should remove only those markers symmetrically opposite those removed by his opponent. Following this strategy, if the first player's opening play is 3, 8, 13 the second player's play should be 18, 23.

But what should the second player do if the opening play is 16 - 18? That's part of the challenge of the problem! Perhaps play 9 - 10, but that seems to increase the first player's chances of making a winning move next time. We do assume the first player is smart even if he does err on his first play. Perhaps play at random, but that seems to decrease the second player's chance of obtaining a winning board configuration.

The complexity of the problem is indeed increased by letting the first player be human. The problem is very good because it is a mini-version of what one often faces in much larger problems: the solution is not trivial; although each step of a solution can be well defined, some definitions will reflect the problem solver's best judgment rather than an absolute truth; once a solution is well defined, a program can be written that plods through many cases while another can be written that uses reflections and rotations of the board to reduce the number of cases. The challenge of writing a program that plays Tac Tix with a smart but fallible user who is given the first move properly belongs under the title "Creative Computing." And those who write such a program are likely to have done some "creative analysis" before they finish.

A modified version of Tac Tix that looks easier but is actually much more complex is played on a 4 x 4 board rather than a 5 x 5 board. The only other change is that the player who removes the last marker is the loser. Is there a winning strategy for either player? After trying to define a winning strategy for one of the players, one may well become interested in writing a program that develops its strategy by learning as it plays. By repeating successful plays and avoiding the repetition of unsuccessful plays, the computer can improve its strategy with each successive game. The writing of such cybernetic programs will be the subject of a future column.

Related References

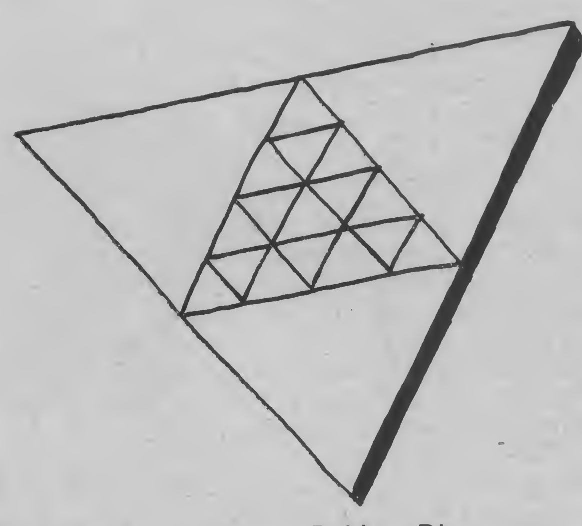
Gardner, Martin; *Mathematical Puzzles and Diversions*; New York: Simon and Schuster; 1959; Chapter 15.

Spaulding, R. E.; "Recreation: Tac Tix"; *The Mathematics Teacher*; Reston, Virginia: National Council of Teachers of Mathematics; November 1973; pages 605-606.

Puzzles and Problems for Fun

This puzzle is calculated to test your ability in calculus: A watchdog is tied to the outside wall of a round building 20 feet in diameter. If the dog's chain is long enough to wind halfway around the building, how large an area can the watchdog patrol?

A. G. Canne
Pittsburgh, Pa.



The Sheik of Abba Dabba Dhu wears this medallion, on which each equilateral triangle represents a wife in his harm. How many wives does the sheik have?

David Lydy
Cincinnati, Ohio

A new computer game from CREATIVE COMPUTING Magazine......

DEPTH CHARGE

DESCRIPTION

In this program, you are captain of the destroyer, USS Digital. An enemy submarine has been causing trouble and your mission is to destroy it. You may select the size of the "cube" of water you wish to search in. The computer then determines how many depth charges you get to destroy the submarine.

Each depth charge is exploded by you specifying a trio of numbers; the first two are the surface coordinates, the third is the depth. After each depth charge, your sonar observer will tell you where the explosion was relative to the submarine.

PROGRAM AUTHOR

Dana Noftle (Age 18)
37 Mohawk Drive
Acton, MA 01720

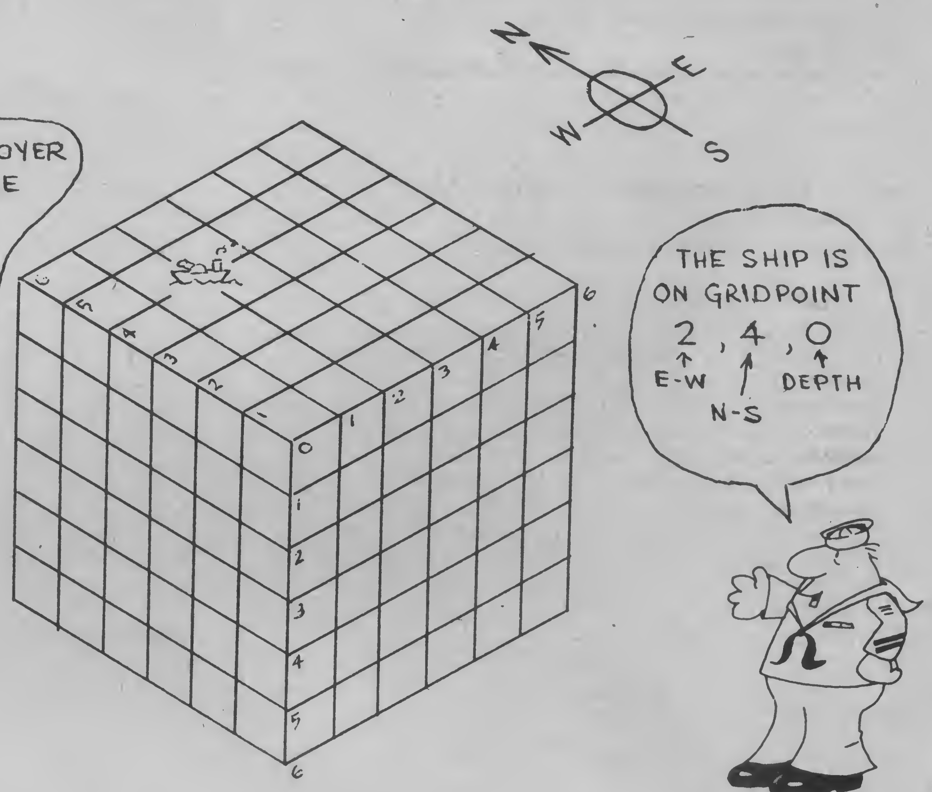
USING THE PROGRAM

1. Type in the DEPTH CHARGE program on your computer. Convert it, if necessary, to your dialect of BASIC.
2. Divide into teams of 2 or 3 players and play the game. Try to come up with an optimal guessing strategy for a search area with a dimension of 10, of 100, of 1000.
3. Statement 30 sets the maximum number of trials allowed for search areas with different dimensions. Make a table like this:

SEARCH AREA SIZE	TRIALS ALLOWED
1	
2	
.	
.	
.	
100	

What does this tell you?

4. Modify the program to allow the submarine to move one grid point in any direction on each trial. It makes a more exciting game, but you'll have to allow additional trials to find the submarine. In this game, make TRIALS N a quantity to be input.



PROGRAM LISTING

```
LISTNH
10 PRINT "DEPTH CHARGE GAME" \ PRINT
20 INPUT "DIMENSION OF SEARCH AREA";G \ PRINT
30 N=INT(LOG(G)/LOG(2))+1 \ RANDOMIZE
40 PRINT "YOU ARE CAPTAIN OF THE DESTROYER USS DIGITAL."
50 PRINT "AN ENEMY SUB HAS BEEN CAUSING YOU TROUBLE; YOUR"
60 PRINT "MISSION IS TO DESTROY IT.  YOU HAVE"N"SHOTS."
70 PRINT "SPECIFY DEPTH CHARGE EXPLOSION POINT WITH A"
80 PRINT "TRIO OF NUMBERS -- THE FIRST TWO ARE THE"
90 PRINT "SURFACE COORDINATES; THE THIRD IS THE DEPTH."
100 PRINT \ PRINT "GOOD LUCK !" \ PRINT
110 A=INT(G*RND) \ B=INT(G*RND) \ C=INT(G*RND)
120 FOR D=1 TO N \ PRINT \ PRINT "TRIAL #"D; \ INPUT X,Y,Z
130 IF ABS(X-A)+ABS(Y-B)+ABS(Z-C)=0 THEN 300
140 GOSUB 500 \ PRINT \ NEXT D
200 PRINT \ PRINT "YOU HAVE BEEN TORPEDOED! ABANDON SHIP!"
210 PRINT "THE SUBMARINE WAS AT"A","B","C \ GOTO 400
300 PRINT \ PRINT "B O O M ! !  YOU FOUND IT IN"D"TRIES!"
400 PRINT \ PRINT \ INPUT "ANOTHER GAME (Y OR N)";A$
410 IF A$="Y" THEN 100
420 PRINT "OK.  HOPE YOU ENJOYED YOURSELF." \ GOTO 600
500 PRINT "SONAR REPORTS SHOT WAS ";
510 IF Y>B THEN PRINT "NORTH";
520 IF Y<B THEN PRINT "SOUTH";
530 IF X>A THEN PRINT "EAST";
540 IF X<A THEN PRINT "WEST";
550 IF Y<>B OR X<>A THEN PRINT " AND";
560 IF Z>C THEN PRINT " TOO LOW."
570 IF Z<C THEN PRINT " TOO HIGH."
580 IF Z=C THEN PRINT " DEPTH OK."
590 RETURN
600 END
```

SAMPLE RUN

```
RUNNH
DEPTH CHARGE GAME

DIMENSION OF SEARCH AREA? 10

YOU ARE CAPTAIN OF THE DESTROYER USS DIGITAL.
AN ENEMY SUB HAS BEEN CAUSING YOU TROUBLE; YOUR
MISSION IS TO DESTROY IT.  YOU HAVE 4 SHOTS.
SPECIFY DEPTH CHARGE EXPLOSION POINT WITH A
TRIO OF NUMBERS -- THE FIRST TWO ARE THE
SURFACE COORDINATES; THE THIRD IS THE DEPTH.

GOOD LUCK


TRIAL # 1 ? 5,5,5
SONAR REPORTS SHOT WAS SOUTHEAST AND TOO HIGH.


TRIAL # 2 ? 3,7,7
SONAR REPORTS SHOT WAS SOUTHEAST AND DEPTH OK.


TRIAL # 3 ? 1,9,7
SONAR REPORTS SHOT WAS NORTHEAST AND DEPTH OK.


TRIAL # 4 ? 0,8,7

B O O M ! !  YOU FOUND IT IN 4 TRIES!
```


Computers Help Watch For Corn Blight

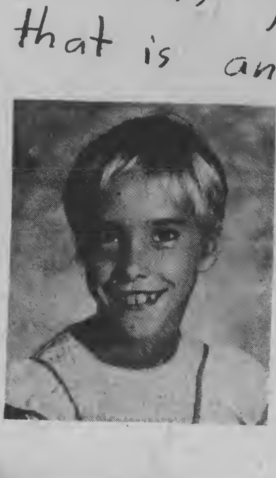
PURDUE UNIVERSITY, LAFAYETTE, IN — Mrs. Susan Schwingendorf, LARS Computer Analyst, marks fields on a photo work copy to assist in locating data in the multispectral analysis. The computer listing indicates the crops grown in each field. Biweekly data from ten corn fields checked for blight by Extension Agents are also made available to her and the other data analysts. The Corn Blight Watch Experiment is being conducted by the U. S. Department of Agriculture, NASA, and the Agricultural Experiment Stations and Extension Services of seven states, in cooperation with Purdue University's Laboratory for Applications of Remote Sensing (LARS) and the University of Michigan's Institute for Science and Technology (IST). (Photo courtesy NASA).

COMPUTERS HELP WATCH FOR CORN BLIGHT

Daddy and his Computer

by Heather Lynne McWilliams

One sunday morning when eveybody except grandmother was dressed ready to go to church and jist then the phone rang daddy had to go to work. So mommy, Alex and I went to church. A few minutes later daddy was back home becuse he forgot his briefcase and daddy was back at work a few mintes later. Mommy, Alex and I came home early becuse Alex thew-up. And when we got back, daddy called he said, "I am comeing home." And when he came home there was a computer chasing his car, he stopcd the car and got out and went in the house and yelled lock the doors shut the window a run away computer he yelled over and over. The news speded fast and soon the police, the police caught the computer and smashed it with a hammer. And daddy quit his job and got another job. But I think that is another story. □



Heather Lynne McWilliams turned in an unusually complete resume after her story was accepted. She is eight years old, four feet tall, weighs 48 pounds, is in excellent health, and is single. She lives in Kailua, Hawaii. Experience: reading books and magazines, such as School Bulletin, Jack and Jill, National Geographic, and Weekly Reader; creative writing since first grade, plus many stories written at home, including "The Haunted House" and "The Singing Gun." Education: preschool at Carey's School; first through third grades at Kailua Elementary School. Outside interests: Swimming, climbing trees, learning to sew, member of church choir, playing the piano, hiking, camping, art, and dolls. Early background: born at Air Force Academy, Colorado Springs, Colo.; lived in Colorado Springs, Montgomery, Ala., and Washington, D.C. Father's job: director of technical services and internal controls for Computing Management Inc.

Contest!

CONTEST RULES

All programs submitted must run in Dartmouth BASIC or FORTRAN IV. Programs running outside these bounds may be submitted, but a fair evaluation cannot be assured. Please submit your entries on paper tape, if written in BASIC; tape or punch cards, if written in FORTRAN. All entries become the property of Creative Computing. They cannot be returned.

The first two or three lines of the listing *and* the output, must contain your name, the name and address of your school, and both your home and school phone number and name and home phone of your teacher.

The winner will head a committee composed of himself, a fellow student and his teacher. Decisions of this committee will be final. The committee will be responsible for establishing the criteria upon which their judgment will be made. Possible criteria are usage of computer technique, uniqueness of the program, form of the program and output, length and running time. Limitations of your system will be considered.

The winner will receive a two-year subscription (or extension) to Creative Computing. Second and third place winners will receive a one-year subscription (or extension).

HERE'S THE PROBLEM . . .

Andrew Kourkoutis defines a perfect digital invariant as an integer containing N digits, where the sum of the Nth powers of the digits is equal to the integer itself. In general,

$\ldots H^N + \ldots + I^N + J^N + K^N =$
$\ldots (10^{N-1}H) \ldots (10^2 I)(10^1 J)(10^0 K)$

If N is 3, then

$I^3 + J^3 + K^3 = 100I + 10J + K$

371 is one such number, for

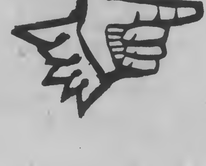
$3^3 + 7^3 + 1^3 =$
$27 + 343 + 1 = 371$

Write a program to find digital invariants for N less than or equal to 4. (If N is greater than 4, running time on most systems will be too great.)

The deadline for entries for this competition is Dec. 20, 1974, according to postmark. For this first contest, we will arrange for the judging. The winner of this contest will head the committee that is to judge our next one.

ANY GOOD IDEAS?

Can you think of a clever poster? Send it along. We'll run it on our computer and publish it here, with your name on it. There is a limit of four lines and seven characters per line. We can also publish computer printed pictures, if you send us finished copy.



Club News

OUR COMPLIMENTS

If your computer club is registered with us, we'll provide a $1.00 discount on personal subscriptions to club members.

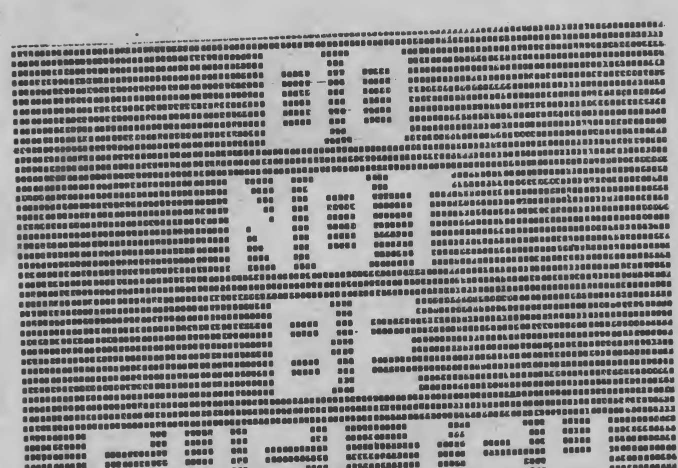
To register your club, send send us a copy of your constitution, the address of your school and a list of names of your members. Include the name and home phone number of your president and sponsoring teacher.

A CHALLENGE

We, the Compunuts of Sylmar High School, offer an open challenge to any computer club to duel at ? miles in BASIC or FORTRAN on the problem of your choice. However, we must warn you that your efforts will be futile. We will vanquish all takers! Our triumph will be absolute!

RICHARD ACUNA
President, The Compunuts
Sylmar High School
Sylmar, California

OUR COMPUTERIZED telephone answering service is scientifically calibrated to judge human personality just by voice vibrations and tonal quality. We can detect within 10 seconds whether the person who wishes to speak to you is trying to borrow money or sell you worthless merchandise. Write for information. Electronic Voice Analysis.



Can You Solve The Energy Crisis?

Dr. Dan Klassen
Computer Technology Program
Northwest Regional Educational Laboratory
Portland, Oregon 97204

Among the many attractions available to EXPO '74 visitors in Spokane, Washington is a computer simulation of the energy crisis. Located in the Environmental Symposia Center, the energy simulation exhibit is designed to allow anyone to step up to an interactive CRT terminal linked to a PDP 8/M and explore various solutions to the energy shortage.

The simulation is used almost continually from 10 a.m. when the exhibits open until 10 p.m. when the gates close, according to Chris Fromhold, an Environmental Symposia Center staff member. A typical user stays at the terminal about 15 minutes and manipulates the model on a year-by-year basis for a simulated 10- to 15-year period. Center staff members report that most users have never used a computer before.

The energy simulation model itself reflects energy supplies and use in the United States. After a brief introduction on how to make changes in the model, the user attempts to balance the demand for energy with available supplies by changing policies, standards and energy use habits. Factors which can be manipulated include: production level of American industries, factories and utilities; pollution standards that plants must meet; efficiency of the automobile as a means of transportation; condition and availability of mass transportation; and use of energy for heating and lighting. The user adjusts energy consumption in each of these areas in an attempt to keep demand and supply in balance. In addition, the user attempts to keep indices of pollution generation, economic well-being and general citizen satisfaction within acceptable limits. Changes to the energy use rates are made on a yearly basis. If the user elects not to adjust energy use patterns, the demand for energy grows at a predetermined rate.

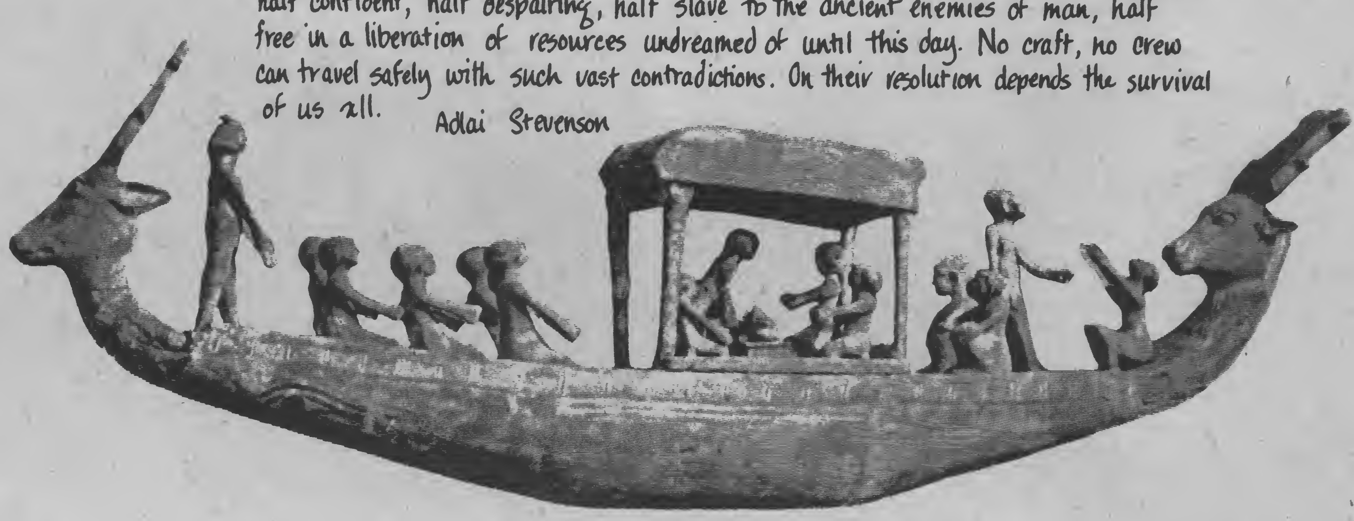
Most of the users are junior and senior high school boys, although many school-age girls as well as numerous older people try the simulation. Fromhold, a teacher during the school year, says that the most interested and excited users are 12- to. 14-year old boys that have never used a computer before.

Staff members have observed that a significant number of users return to the exhibit with a strategy in mind. Of these, the most troublesome are junior and senior high school boys who determine how the program operates and then use this knowledge to introduce a "bug" into the system; debugging is necessary about once every week, according to Symposia staff members.

The exhibit was created and produced by Dr. Dan Klassen, John Lynch, Sr., and Dr. Duane Richardson of the Computer Technology Program, Northwest Regional Educational Laboratory, Portland, Oregon. The energy simulation was originally designed, under a grant from the United States Office of Education, as an instructional application for classroom use. It is one of five energy-related, computer-based instructional applications being developed by the Computer Technology Program; all will be available in early 1975.

We travel together, passengers on a little spaceship, dependent on its vulnerable reserves of air and soil; all committed for our safety to its security and peace; preserved from annihilation only by the care, the work, and, I will say, the love we give our fragile craft. We cannot maintain it half fortunate, half miserable, half confident, half despairing, half slave to the ancient enemies of man, half free in a liberation of resources undreamed of until this day. No craft, no crew can travel safely with such vast contradictions. On their resolution depends the survival of us all.

Adlai Stevenson



Let's Take A Look At . . .

Pythagorean Triples

by Richard Acuna, Senior
Sylmar High School, California

PROBLEM:

A Pythagorean triple is a set of three positive integers that satisfy $A^2 + B^2 = C^2$. Let's find some of these triples.

PROCEDURE:

*On a piece of scratch paper, write out a table of squares of integers, from 1 to 15.

*Assume that A is less than or equal to B. Then the smallest value for A or B would be 1. Question: Is $A^2 + B^2$ a perfect square, when A and B are equal to 1?

*$1^2 + 1^2 = 2$. Since 2 is not a perfect square, this combination for A and B does not produce a Pythagorean triple. Make a table showing all possible combinations of A and B, if each is less than or equal to 10. Compare 1 with 1, 2, 3, . . ., 10, then 2 with 1 to 10, and finally, 10 with 1 to 10. Make note of all the triples you find. Use the table of squares you made, to check the sum of $A^2 + B^2$ to see if it is a perfect square.

*If you were asked to find all triples in which A and B were less than 25, using only a pencil and paper, the task would take a long time. Write a program for your computer to find all triples in which A and B are less than or equal to 25. How long did it take your program to find these triples? How long would it have taken, with only pencil and paper?

SOME QUESTIONS:

*Count the number of sets of triples you found above. If A and B range between 25 and 50, how many triples would you expect to find? If A and B range between 50 and 100, how many triples would you expect to find? Modify the program you have written, to find all triples for A and B less than or equal to 100. Check your answers to the questions above. Does it appear that, as A and B range from 1 to 100, the Pythagorean triples are evenly distributed?

*If you study the set of triples you have found, you will note that some triples are multiples of others. For example, 6, 8, 10 are multiples of 3, 4, 5. If these integers are measures of sides of right triangles, both triangles would have the same shape. Modify your program, so that it will output only those triples that would relate to uniquely shaped triangles; eliminate all multiples of any triple found.

*Given: $A = 2MN$, $B = M^2 - N^2$ and $C = M^2 + N^2$. These formulas give Pythagorean triples whenever M and N are integers. Use these formulas and a new program to find a set of triples. Do these formulas eliminate multiples? Did you find more triples with this program? Did your program overlook any triples?

*If you have the computer time, use one of your programs, modified, to find all triples for which A and B are less than 10000. Does it appear that triples are evenly distributed throughout this range?

Speak Your Piece!

Tax time rolls around. The IRS throws out your tax return, claiming you owe an additional $1000. You complain, of course, and after weeks of anxious waiting, the reply may come back; computer error.

I'm sick and tired of people using the computer as a scapegoat. Somebody should tell somebody that a computer can only do what it is told to do by people.

Larry Troth, Sophomore
California State Univ.
Northridge, CA

> "If he is indeed wise he does not bid you enter the house of his wisdom, but rather leads you to the threshold of your own mind."
>
> Kahlil Gibram

Poster

Put the poster on the next page up on the bulletin board for all to enjoy. Please *don't* use the subscription form on the poster – you'll find another one in the back of the magazine.

Watch for a new poster every issue of CREATIVE COMPUTING – and from now on they'll deal with computing!

creative computing

★ a lively new non-profit magazine of educational and recreational computing ★

the editors discuss the cover design for the latest issue of CREATIVE COMPUTING

Hewlett-Packard's Computer Curriculum Project

Hewlett-Packard Company publishes several series of curriculum material which cover secondary through graduate level education. The material assumes that students have the use of the BASIC language computer system. Monitoring the project is a group of educators organized into an advisory board. This group reviews all the materials and makes recommendations for further development. The authors of the material are educators — well known for their work in computer curriculum and for accomplishments in their academic fields as well.

Books are available now for Secondary School use in selected topics in mathematics, physics and ecology. A number of Social Science units will be introduced for the 1974-75 school year. Each curriculum unit covers a standard or enrichment topic. The computer is used to develop concepts which are either impossible or very difficult to explore without the computer, and which make a significant contribution to a fuller understanding of the subject. Most modules consist of student and teacher's versions. The student version emphasizes computer-oriented presentation of the material and gives exercises for students to do. The exercises are open-ended, in order to challenge each student to work at his or her own level of ability.

The teacher's advisor contains background material, suggestions for presentations, and sample solutions for student problems.

If special computer programs are required, the listings are included in the material making it unnecessary to purchase them separately.

The following list contains the titles which are now available for Secondary School use. For further information or to be placed on a mailing list please write:

Computer Curriculum Project
Hewlett-Packard Company
11000 Wolfe Road
Cupertino, CA. 95014

MATHEMATICS

Attacking Non-Linear Equations Student Text
Attacking Non-Linear Equations Teachers Notes
Number Sets Student Lab Book
Number Sets Teachers Advisor
Mathematical Systems Student Lab Book
Mathematical Systems Teachers Advisor
Functions Student Lab Book
Functions Teachers Advisor
Linear Equations & Systems Student Lab Book
Linear Equations & Systems Teachers Advisor

PHYSICS

Geometrical Optics Student Lab Book
Geometrical Optics Teachers Advisor
Mechanics Student Lab Book
Mechanics Teachers Advisor
Waves Student Lab Book
Waves Teacher Advisor
Electricity & Magnetism Student Lab Book
Electricity & Magnetism Teachers Advisor

ECOLOGY

GRAZE Ecology Simulation Student Text
GRAZE Ecology Simulation Teachers Notes
Air Pollution Student Lab Book
Air Pollution Teachers Advisor

PROJECT SOLO COMPUTER TOPICS

Trigonometry Student & Teacher
Mathematics Projects Student Text
Mathematics Projects Teachers Guide
Calculus Student & Teacher
Matrix Mathematics Student Text
Matrix Mathematics Teachers Guide
Physics Student Text
Physics Teachers Guide

Selected books from this project will be reviewed in CREATIVE COMPUTING in forthcoming issues. Check the Book Review Section.

NOTONE -- A Challenge!

Something to ponder--. Determine the BEST strategy for playing the following game. If you think you have a good one, send it to Walter Koetke at Lexington High School, Lexington, Mass. 02173. If requested, he'll send you a punched paper tape of a program that plays the game reasonably well (not best). Tapes will only be sent to those submitting solutions with their request.

The game NOTONE is played with two players and a pair of dice. There are ten rounds in the game — one round consisting of one turn for each player. Players add the score they attain on each round, and the player with the highest score after ten rounds is the winner.

On each turn the player may roll the two dice from 1 to N times. If T_i is the total of the dice on the i^{th} roll, then the player's score for the turn is $T_1 + T_2 + T_3 + \ldots + T_N$. However, and here's the catch, if any T_i is equal to T_1 then the turn is over and his score is 0 for that turn. After each roll that doesn't equal T_1, the player can decide whether to roll again or stop and score the number of points already obtained.

Creative Computing will print the best game submitted next issue.

NORTHWESTERN UNIVERSITY

EVANSTON, ILLINOIS 60201

COMPUTERS AND TEACHING
2003 SHERIDAN ROAD

312 492-5367

May, 1974

The Computer Aids to Teaching project is now organizing a "task-force" to work with CONDUIT on the problems of transferability of "hard-core" Computer-Aided-Instruction lessons (or programs, if you prefer). By "hard-core" we mean computer based instruction in which a language specifically designed for C.A.I. is utilized (Coursewriter, Planit, Tutor, etc.). To gather as representative a group as possible, we would like to ask you to circulate the enclosed materials to people you know would be interested in the transferability problem. (We can supply you with copies for a mailing, if you like.) CONDUIT, till now, has been concerned with many types of instructional computer programs, but has not gotten into "hard-core" C.A.I., and it is this area with which our task-force will concern itself. The task-force will probably not be meeting as a group, but will communicate through the Postal Service (and for those of you who can, through a computerized conference on Northwestern's computer system). If you are willing to be an active participant in this project, please return the enclosed form with your comments.

Notes on CONDUIT Documentation Guidelines (enclosed):

We suspect that you will find that some of the points in the abstracted guideline either do not apply or are not specific enough for C.A.I., and we would appreciate it if you would mark changes or additions on the Guideline or a separate sheet of paper and send it to us. In reviewing the guideline, please consider the viewpoints of either the authors of C.A.I. lessons, or the users of C.A.I. lessons, and indicate with which points it would be most difficult to comply (in the author's case) or most necessary (in the user's case). We know the guideline is difficult to live up to, but we feel it is designed to help the user evaluate a software product in advance, and then to implement it in as little time as possible, once adopted.

Task-force participants -- CONDUIT "hard-core" C.A.I. transfer

name

address

phone

"hard-core" systems you have used:

type of computer you use or have used:

number of students using C.A.I. at your site:

number of programs (lessons) you think could be added to a potential CONDUIT test repository by your site(using Guideline):

CONDUIT DOCUMENTATION GUIDELINE
Abstracted Version

CONDUIT: A consortium of regional networks at Oregon State University, North Carolina Educational Computing Service, Dartmouth College, and the Universities of Iowa and Texas (Austin).

INTRODUCTION

The CONDUIT Documentation Guidelines describe standards for packaging computer-based curriculum materials. These guidelines stipulate those types of documentation deemed necessary not only for local usage but also for transportation of packages, both educationally and technically, among multiple non-homogeneous computer installations. Documentation to comprise a complete package will be collected from appropriate sources serving the following purposes:

DOCUMENTATION CHART

DOCUMENT	*SOURCE*	*TARGET*	*PURPOSE*
1. COVER LETTER	Local Curriculum Coordinator*	Faculty User	Availability announcement
2. ABSTRACT	Author	Reviewer or Discipline Committee { Faculty User	Preview
3. EDUCATIONAL DOCUMENTATION	Author	Faculty User, Student	Substantive & Pedagogical Aspects
4. TECHNICAL DOCUMENTATION	Programmer, Author	Curriculum Coordinator Faculty User, Student	Technical Aspects (systems & programming considerations)
5. LOCAL DOCUMENTATION (educational & technical)	Local Curriculum Coordinator	Faculty User, Student	Modifications resulting from usage at a specific network computer center
6. REVIEWS	Reviewers, Discipline Committee**	Faculty Users	Certification

NOTE: This abstracted version was prepared by Trinka Dunnagan from the *CONDUIT Documentation Guidelines* of October, 1973 in which each component of documentation is explained more fully.

*Each local Curriculum Coordinator is responsible to CONDUIT for collecting and disseminating computer-related curriculum material at his network schools.

**Discipline committees, who serve in an advisory capacity to CONDUIT, are composed of faculty with interest and expertise in computer applications within a discipline.

1. COVER LETTER

A *cover letter* should be supplied by the local Curriculum Coordinator as supplementary, local documentation for any CONDUIT package. The purpose of such a letter is to provide prospective network users with a realistic picture of their relationship with CONDUIT in the event of the package's adoption and to state the costs, availability, and procedures for acquiring any materials (texts, card decks, magnetic tapes, program listings, etc.) needed for that adoption. Any discussion of availability should list in addition to the source repository other network centers and computer types at which this package is supported. Any restrictions (such as copyrights, etc.) on the reproduction and use of these materials should be stated at this time as well as the caveat that although CONDUIT has attempted to maximize the educational value and technical reliability of the programs, *no liability* for errors in these materials, expressed or implied, is assumed by the author(s), the source facility (NAME), or the CONDUIT consortium. User's expectations of CONDUIT support (computer time, programming aid, consulting services, etc.) should be specifically covered in terms of extent, personnel and mechanics involved.

2. ABSTRACT

The author of each package should provide an abstract sufficient to serve as an all purpose introduction to the computer-based curriculum package. The abstract should contain the following items:

(a) Descriptive title
(b) Mnemonic or calling name
(c) Author(s) and original source
(d) Names and locations of any subsequent modifiers
(e) Summary of substantive content
(f) Statement of educational objectives
(g) Specifications of computer's instructional role
(h) Background requirements for instructors and students
(i) References (if available)

3. EDUCATIONAL DOCUMENTATION

The author of a computer-based curriculum package addresses in this type of documentation the needs of faculty and student users. Topics covered should include the following:

(a) Substantive aspects—theoretical background and disciplinary principles of this computer-based package.
(b) Pedagogical significance—relationship of the educational objectives stated in this material to content and instructional techniques.
(c) Implementation considerations—suggestions for instructional management within the classroom, the computing environment and the standard curriculum.

These topics could appear in separate single-purpose manuals (e.g., Teacher's Guide, Student Manual, Programmer's Guide, Problems, etc.) or as a collective write-up depending on the size and scope of the package being documented.

4. TECHNICAL DOCUMENTATION

Technical documentation involves those materials necessary for the understanding by a potential user of how the courseware (e.g., program) operates. Technical documentation should attempt to extricate that programming documentation which is universal and transportable, in terms of information content, from site-specific descriptions needed only for local usage. In order to clarify both the program logic and function, one needs:

(a) Explanation of the program's logical organization and the functions of any discrete modules.
(b) Well-commented program listing.
(c) Descriptions of sample input such as test data and any program parameter cards required.
(d) Listing of output generated by such sample input.
(e) Test problems and expected results.
(f) Description of program options and how to exercise them.
(g) Listing and explanation of program-generated messages.
(h) Glossary of variable names and special discipline oriented items.
(i) Itemization and descriptions of supporting programs, subroutines and external files.
(j) Formats for parameter and data cards.

For each data set, the following information should be available:

(a) Description of file organization.
(b) File data item definitions.
(c) Indices to the data set.
(d) Security procedures, if any.

5. LOCAL DOCUMENTATION

Curriculum Coordinators should provide network users with instructions for usage of CONDUIT materials at their local computer center. Information classifiable as local (or site-dependent) is necessary only if it differs from that which is provided as universal documentation. Such information includes:

(a) Instructions and examples on how to use the local computer (JCL, deck setup, typical file or program access, major operations.
(b) A setup example of how to access the program under consideration.
(c) A listing of the local version of the program and notations on its significant differences from the universal model.
(d) Sample I/O for this version of the program.
(e) Information on typical run times and related costs.

In addition an overall description in explicit terms of the current computing environment should state:

(a) Operating system, release level/version.
(b) Mode of usage (batch versus interactive).
(c) Storage devices required.
(d) Special peripheral device requirements (central site).
(e) Special terminal needs (user site).
(f) Common causes of program failure and error recovery.
(g) Any other locally pertinent information or comments on these materials.

6. REVIEWS

Material selected for dissemination through CONDUIT is first reviewed under the direction of an appropriate discipline committee. Those packages receiving positive reviews are included in the CONDUIT Library for undergraduate education with solicited reviews comprising a final section of the documentations.

Confess: A Humanistic, Diagnostic-Prescriptive Computer Program to Decrease Person to Person Interaction Time During Confession

KENNETH MAJER
Institute for Child Study

MICHAEL C. FLANIGAN
School of Education
Indiana University

Recent Vatican interest in the effect upon laymen of the shortage of professional priests (PP) and the decreased seminary enrollment of potential priests (P'P) has led to the development of Computerized Operations (Non-retrievable) for Expediting Sinner Services (CONFESS). This program provides a viable alternative to traditional confession procedures by listing penance requirements (by sin) on a private print-out to confessees appropriate to the sin committed. This eliminates one problem which frequently occurs where the confessee, because he is under extreme duress, may forget the original penance. In addition, the program provides a probability estimate of the consequence of not completing the penance associated with a given sin; for example, number of years in purgatory. Thus, full freedom of choice is given to the participant/user (PU). The program requires no PP involvement and hence frees PPs to engage in more pressing activities. It is hoped that by providing PPs with more time for critical theological activities, P'Ps will consider the priesthood a more socially conscious and relevant profession, causing an increase of P'P enrollment in accredited seminaries.

Program Description

CONFESS is available in three natural interactive languages, COURSE WRITER III, BASIC and TUTOR and can be programmed for most other natural languages such as interactive FORTRAN. The program has been developed utilizing on-line computer terminals linked to an IBM 360 for data input, but could be modified to operate in batch mode on almost any third generation configuration given the willingness to sacrifice immediate feedback.

The computing procedures for CONFESS are as follows: The present sins input (psi) yields the graduated penance accrual (GPA) as a function of present sins (ps) plus frequency of confession visits (fcv) times completed penances (cp) divided by recurring sins (rs). Hence, GPA is a function not only of the immediate sins reported but also a partial function of the reciprocal relationship of recurring sins to completed penances by frequency of confession visits. The relative penance, then, is increased by the inclusion of recurring sins.[1] Mathematically, this can be represented as follows:

$$psi \rightarrow \quad GPA = f \quad \left\{ ps + fcv\left(\frac{cp}{rs}\right) \right\}$$

Therefore, each present sin yields a specific GPA that is stored until all GPAs have been computed. At that time, punishment and its maximum likelihood of occurrence[2] should the GPA not be completed, are retrieved from core storage and printed out for the individual GPA prescription.

Validity and Reliability

A study to establish the validity of the CONFESS program was conducted. The procedure included a sample of 243 actual confessions stratified across low, medium and high socio-economic income brackets with non-significant differences in proportions of black, white and Spanish speaking PUs. Fourteen priests were used in the study from seven different cities.

The actual sins confessed and penances prescribed in the confessional booths were tape recorded without the confessor or confessee's knowledge to insure absolute authenticity of confessor-confessee interaction.[3] The tapes were further analyzed and penances were rated on a scale of 1-10 where 10 = maximum severity.[4] Then ratings were made by the seven cardinal evaluators identified by Stake (AERA, 1972). The interrater reliability was .949.

The 243 sin sets taken from the taped confessions were then entered into the CONFESS program via remote terminal. A Pearson product moment correlation was computed between the actual PP penance prescriptions and the CONFESS PGAs. A correlation of .971 was interpreted to provide sufficient concurrent validity for CONFESS confidence.

A further series of small studies to determine the reliability of the CONFESS program were conducted as follows:

Study 1: External Latency Reliability. The mean wait for confessional booths with PPs (where there were 2 booths/church) was 7.12 minutes while, in comparison, the average wait for a CONFESS box (one installation per church) was only 1.72 minutes. This difference in out-side wait latency is significant at the $p < .01$ level.

Study II: Internal Latency Reliability. This study examined the latency from the last sin confessed until the PP or CONFESS program provided the penance or GPA, respectively. Again, the CONFESS latency was significantly shorter than the PP latency. The means were 1.31 minutes (plus an average of 9.3 head shakes) for PPs, and 6.1 seconds for CONFESS.

Study III: Computer Breakdowns vs. PP Rest Breaks. In this study the CONFESS program was monitored for computer breakdowns and don't-understand-not-compute-either (DUNCE) loops. During the 243 CONFESS program runs (a total of 517 minutes), no breakdowns were reported and only one (1) DUNCE loop was reported. The DUNCE loop was in the case of one PU who was previously excommunicated from the church; however, the CONFESS program has been modified and will now process excommunicated PUs as well as non-excommunicated PUs. PPs, on the other hand, showed an average of 1 rest break for a mean of 12.3 minutes every hour and one-half.

Study IV: Consistency of PP penance vs. GPA. In this study, the 243 confession tapes were re-heard by the same 14 PPs. Each PP re-heard the same confessee's albeit on tape and without hearing the end of the tape which contained the penance he gave. In 241 cases, the PPs did *not* give the same penance and, in fact, in 191 cases the penance severity changed at least one degree (e.g., from a severity rating of 7 to a severity rating of 8). Although no speculation for causality is made here, it is important to compare the CONFESS consistency. In all 243 cases, the GPA was identical.

The results of these four studies are sufficient to provide confidence in CONFESS program reliability.

Procedures/Output

Being a natural language program, the procedures for CONFESS are extremely simple. The following steps describe the PU procedures.

Step 1: Enter the CONFESS box,[5] and kneel on cushioned kneeler in front of the typewriter/console. Type in your personal PU identification code.

Step 2: The typewriter will type your name and the elapsed time since your last CONFESS session (CONFESSION). Following the request for present sins, type in all sins since your last CONFESSION.

Step 3: Press the "enter" button and silently repeat the short form of the ACT of Contrition. (Given the average latency for GPA, 6.1 seconds, this is usually reduced to "I'm sorry").

Step 4: Remove the CONFESS personalized GPA printout.

Sample Printout

CONFESS GPA PRINTOUT JOHN POPE Age 29

TIME SINCE LAST CONFESSION = 3 WEEKS

PRESENT SINS	TYPE	GPA	PUNISHMENT	PROBABILITY THEREOF
1. SECRETLY ENVIES BOSS	VENIAL	10 OUR FATHERS. PRACTICE SMILING AT BOSS	1 YEAR IN PURGATORY	.98
2. SWEAR AT WIFE	VENIAL	10 HAIL MARY'S. PRACTICE SMILING AT WIFE	1.73 YEARS IN PURGATORY	.84
3. COVET NEIGHBOR'S WIFE	MORTAL	ONE ROSARY/DAY FOR ONE WEEK. PRACTICE SMILING AT WIFE.	ETERNAL DAMNATION	.91

ONLY *3* SINS THIS TIME *MR. POPE.* YOU'RE IMPROVING. YOU HAD *14* LAST CONFESSION. NICE GOING. KEEP UP THE GOOD WORK. LET'S SEE IF YOU CAN MAKE OUT A LITTLE BETTER WITH NUMBER *3* IN THE FUTURE.

PLANIT: The Portable CAI System

by Charles H. Frye*

The first prototype of PLANIT was put into operation early in 1966. With it one could author and dispense typical CAI lesson senarios with particularly good facility in numerically-oritented lesson materials. Five years ago the PLANIT group was proposing a machine independent operating system for CAI that could easily be installed on any garden variety computer hardware and be used to implement the PLANIT language in time-shared fashion for authoring and dispensing CAI. In contrast to the prototype PLANIT, the new portable version was to be a complete system including all the necessary pieces for production CAI. To include only a lesson authoring and execution capability would hardly qualify to be called a system in the terminology of most computer centers. They also want such capabilities as file maintenance and backup, program maintenance, user accounting, device allocation, etc. Thus, PLANIT is much more than just a CAI language.

Confidence was lacking at the outset that the development of a truly portable CAI system could be achieved. One widely respected systems expert gave the project less than ten percent chance of success. Others said that PLANIT would not be alive at the end of the five-year period. It is to the credit of the National Science Foundation that they continued to believe that the goal was attainable. Today, the proposed PLANIT system has been completed and is being tested and used at a number of sites. PLANIT has been installed on a wide variety of hardware, including CDC 3170 3300 & 6500, DEC 10, Honeywell 200, IBM 360/40, 360/67 & 370/155, Siemens 4004 & 155, Telefunken, Univac 1108, and XDS 940, some under existing time-sharing systems and others under batch. One version runs side-by-side with a spooling system, HASP.

It was promised that PLANIT would run 20 terminals in no more than 256,000 bytes of core – it is now running more in less space. In at least one case it ran 12 terminals in 72,000 bytes of core. Being completely modularized, it can be configured to the core that is available.

The installation cost was promised to be in the range of $10,000 to $20,000. Purdue University recently completed a study in which PLANIT was installed on their 6500 and run for a week long pilot study during which 56 PLANIT users ran an average of 49.8 minutes each. They reported a total installation cost of about $1300 requiring about 147 hours of programmer time. Others have also confirmed that the original cost estimates for installation were unnecessarily high.

On the question of consumptiveness, Purdue reported several statistics based on their pilot study:

1. Throughput was apparently unaffected by the operation of PLANIT. Average jobs per hour for the week of pilot study was 434 compared to an average of 421 for the week prior and 445 for the week following.
2. PLANIT required approximately 1/5 of available core while it was processing a user, consuming an average of 4.15 seconds of CPU time per 50-minute period (reflecting the expected low CPU usage for CAI).
3. Terminal/hour costs ranged from $2.08 to $2.25, using Purdue's standard charging algorithm.

It has been observed that because PLANIT uses FORTRAN in the installation process, the result must surely be inefficient especially since such a simple subset is used to implement complex system functions such as scheduling and cataloguing. While it is true that machine language would run faster, two things should be kept in mind:

1. Contrary to all other transfer methods short of recoding, PLANIT execution efficiency will not suffer when the system is moved from one computer to another.
2. Typical CPU usage is very low for CAI such that small differences in efficiency will have only marginal effect. There is no efficiency loss in use of space and peripherals where the CAI investment is high.

At one site in particular there is little doubt that PLANIT is being used as a production system – the University of Freiburg in West Germany has been using it in that fashion for more than a year now with a daily operating schedule, courses being taught for credit and authors hired full-time. Being first, they experienced more than their share of problems but they report the system has operated quite reliably for the last several months.

The PLANIT system is an interpreter – intentionally so. This means that lesson material is stored internally in original typed (or keypunched) form and the keystroke characters are deciphered on-the-fly as the lesson is dispensed to students. There are those who talk about interpreters as though their existance is justified only because the developer didn't know how to build a compiler. In the event that this opinion might still persist, consider the following as some advantages of an interpreter:

1. Interpreting enables certain options in the language which would be difficult or impossible to compile and, since CAI is normally concerned with a user community who is new

*Dr. Frye is the originator of the PLANIT system and is currently project director of the PLANT Project at the Northwest Regional Educational Laboratory, 710 SW Second Ave., Portland, OR 97204.

to computing, the concept of compiling prior to execution will be a new one. Compiling tends to encourage the adoption of language conventions aimed at easing the compilation task at a cost of user convenience. For example, many languages use "counter" items to code user response paths because they can more easily be used in later decision points. Many aspects of PLANIT take on a definition only after the student responds (too late for efficient compilation).

2. Compute times are usually greater for interpreted programs than for compiled programs, ranging from a little greater for character shuffling to more than a hundred times greater for number crunching. However, much of CAI is character shuffling and compute times are characteristically small even for interpreters as was evidenced in the Purdue statistics.
3. Space will usually be of more concern in CAI applications than time. Since a compiled program grows in size according to the number of source statements while an interpreted program usually operates in a fixed size, there will be a point beyond which the compiled program will be the largest. Source code is normally more compact than compiled code. CAI programs are typically large (as programs go) and will pass that point very quickly. Therefore, interpretive CAI systems will ordinarily occupy less total space than compiling ones – this added to the fact that most compiling systems also retain the original source code for editing purposes.

In general, space will probably be more costly to CAI than compute time and interpreting systems will normally require substantially less space. The PLATO terminal has the microfiche projection capability to alleviate this problem but raising the preparation costs somewhat.

Let me now turn to the second part and offer some comments that attempt to put PLANIT in perspective with two well-known elegant CAI systems, PLATO and TICCIT. Today's CAI users are fortunate to have them as options. PLANIT is also an option. The three systems can be summarized in this way:

1. PLATO is for the person who can have everything and has the money to pay for it.
2. TICCIT is for the person who has nothing and wants a lot but has little money to pay for it.
3. PLANIT is for the person who has equipment and needs to get along on what he has because he has virtually no extra money for CAI.

For some time PLANIT was considered to be an interim system, to be used until PLATO and TICCIT were ready. Support required for PLANIT has been a fraction of that for the other two. No hardware development efforts were involved since PLANIT is completely software, at least up until the time of installation. Thus, PLANIT was less expensive to develop and more quickly delivered, making it a likely candidate for interim needs. However, it is becoming increasingly apparent that PLANIT's portability is providing an option not yet available in another CAI system. PLANIT can be mounted on existing equipment with little or no extra hardware investment. PLANIT lessons on all such installations are fully compatible and can be exchanged freely. Experience at operating versions of PLANIT which have been installed on widely differing hardware show no discernable differences to the user. It is a striking experience to sit at the keyboard of a strange system where PLANIT is mounted and immediately be completely familiar with the entire operation. This has been particularly useful to the military with their diversity of hardware. The ability to produce a fully compatible system on existing hardware at nominal cost suggests something more than an interim role for PLANIT. It may well continue to be a viable option for some time to come.

When discussing PLANIT's future, the question is inevitably asked, "Can PLANIT handle graphics?" The answer is "yes" but the implementation of that is probably 95 percent installation hardware and software and 5 percent PLANIT additions. "Holes" have been intentionally left in PLANIT's command structure to allow for these kinds of additions. In one experiment with graphics, a Rand tablet was used for a PLANIT terminal with the display projected onto the under side, giving the impression of "inking" a surface with an electronic pen. It has also been observed that the PLATO plasma terminals would make nice PLANIT terminals. These kinds of questions are decided at installation time.

Probably the most significant recent development in CAI is that we are beginning to have some options – not just the name of the language but the kind of system. Formerly, we had only one option – invent our own unique system. Now there are several more, such as PLATO, TICCIT, and PLANIT. If this progress is to be sustained, then interested parties should feel obliged to see what is available before concluding that nothing currently exists. Articles on CAI appearing within only the past six months in respected national periodicals show that this has not yet occurred. One lists PLANIT and TUTOR (PLATO's author language) among others as "large and complicated and troublesome to learn" and then proceeds to describe a language of dubious improvement bound to specific hardware. Another describes yet a different language in which the technique of prompting the author for lesson inputs is ostensibly "discovered," not mentioning that PLANIT has been doing this for eight years in addition to several others that can also be named. It was especially interesting that after discussing the remarkable gains in efficiency which were attributed to prompting, the authors of the article then proceeded to describe the soon-to-be-released version II which will allow batch input.

What we need are real, legitimate options in CAI, not contrived ones. PLANIT is one of these legitimate options.

The Parable of the Horse

Being a diversion intended to produce a few horse laughs and perhaps some horse sense on a subject that has already fertilized too many fields.

by
John M. Nevison

Author note:

The author has wrangled programs of his own for many years. He has served as a shipping clerk for Project Conduit at the Kiewit Computation Center of Dartmouth College. Presently he is engaged in some private ranching on forest watershed models.

The Parable

There was a farmer who caught a wild horse. He put the animal to work, enlarged his farm and called it a ranch. The idea caught on and soon everyone was after horses to help with their work. Horsemanship developed apace. The wrangler was born. Horse trading flourished. So did stealing.

After the railroads came, horses were shipped from one range to another. Ranchers benefitted from the increase of good animals. And the many regions prospered as the artistry of their breeding and the skill of their training improved the quality of their stock.

The Animal

Bad tempered programs kick even their own handlers.

A computer program has much in common with a wild horse. It is temperamental; docile one moment and vicious the next. With good training it will return a lifetime of hard work. It usually performs well for only one person. And like all living things, it eventually grows old and dies.

Programs and horses exhibit a similar variety. There are frisky colts, hard racing two-year-olds, even tempered mares, ornery stallions, heavy hooved plow-horses, high stepping trotters, quick turning quarter-horses, and broken down nags.

The Home Range

A professional who uses a computer program without knowing how to write one is like a rancher who has never ridden a horse: he may succeed, but he does so under a severe handicap.

Often a skilled academic will train a string of programs to cover his range. How he uses a program will vary with the geography of his subject, the landscape of his students and the weather of his research. Different animals are appears in unfamiliar harness, performing inappropriate work.

When shipping clerks from several stations agree on how to immobilize the program in a standard character code, with certain formats for magnetic tape or punched cards, the cost to load, appropriate to different tasks on a constantly changing range.

A teacher who has learned how to ride a program uses it with discrimination. He knows what it does best, what its limits are, and how to use it most efficiently. Further, like the rancher who knows how to ride, he can manage with skill the wranglers in his employ.

The program wrangler often adapts a preexisting program to his own use. Even the most docile of programs will kick and buck when a new wrangler begins to work with it. Very few programs are well mannered enough to work for someone other than their master. Not surprisingly, a wrangler learns to examine what he wants to do and voids difficult retraining by breeding his own program.

To get his ranch work done the professional is constantly deciding whether to have his cowboys breed a horse suitable to the task or import one from another range and train it to the local tasks. The decision depends heavily on the size of the program.

The Size of the Beast

A nag is a nag.

All kinds of programs come in one of three sizes: large, small, or middle. Large ones are those that take months of a programmer's time or tens of millions of bytes of storage on a computer. Small ones are those less than ten lines long. The middle size ones are those left in between.

A very large program is cumbersome enough to move and store that a telephone connection to the source often costs less than moving it.

A very small program is, by definition, the sort of animal anyone can breed with no trouble on his own range.

Programs between these extremes, written in a language like APL, BASIC or Fortran compose the bulk of the herd currently used in research and instruction. Different academics will choose either to breed or to import a middle sized program after examining the local conditions and the cost of shipment.

Shipping a Horse by Train

Moving a program from one computer to another is like shipping a horse by train: fundamentally, it does not want to go.

The cost of moving a computer program is hard to estimate and the problems of shipping are poorly understood.

The cowboy's preconception of how to move one has proved wrong. His solution was to ride the program into the boxcar, slam the door, ship it and tell someone to ride it out at the other end. A program actually shipped this way usually seriously injures the first handler to get near it when it arrives. After inflicting multiple lacerations on the wrangler, it often dies of neglect. Sometimes it ship, and unload a program shrinks dramatically. Unloading time can be cut from forty hours to fifteen minutes.

Further agreement on the use of languages and on the necessary technical directions can speed the preliminary grooming of an imported program.

The results of these agreements make a program cheaper for a teacher to order from another college and easier for him to handle once it arrives on his campus. However, an easy-to-unload program has to have been loaded with care somewhere else. This loading is not only a technical problem, it is a personnel problem.

Rancher, Cowboy, Clerk

Anyone working around a horse must learn to handle a pitchfork.

The teacher, the programmer, and the computer center shipping clerk each view the program from a different perspective. The teacher wants the program to do the job, be it in research or in teaching.

The programmer also wants the program to perform well. However, his preoccupation is with training the program, smoothing its quirks and polishing its routines so that it will perform at its best. He is often beguiled into prizing the elegance of the performance more than the accomplishment of the job.

The shipping clerk knows that the program must be crated with appropriate instructions and shipped in a confined, unnatural state. The cowboy, preoccupied with the program's beauty on the open range, views this clerk with thinly veiled hostility. The clerk, preoccupied with shipping the program well, views the cowboy with ill-concealed scorn. Neither fully appreciates the other's point of view.

The wise academic shares the programmer's love of the program on the open range, but realizes the need for shipment if the program is to get to other ranchers at different stations. He will tolerate the shipping clerk and perhaps even help him.

The clerk will demand of the rancher that he provide a program well trained for use by others. In addition he asks for details on the program that will help his counterpart at another computer center to get it working according to the advertisement. These details include sample runs of the program and notes on the limitations of its intended use.

Working with the program's handler, the clerk will crate the program with appropriate technical directions and ship it.

Station to Station, Range to Range, Rancher to Rancher

It is not the program's fault if it is harnessed to the wrong job.

The advertisement for the program could be a research note, published article, textbook or talk. It is from this advertisement that an alert professional first learns of a new application. Only after he has found out how he might use a program does he decide to import a copy of the original.

The rancher comes down to the station to meet the train when a program arrives. He and the clerk discuss how the program should be groomed at the station and settle on a handler. The handler may be the teacher himself, one of his students, or a center employee. The wrangler will curry and brush the program and make sure that its hooves are properly shod for the region. He may help the rancher lead it home to see how it performs.

During this final trial the rancher is preoccupied with checking advertisement against actual performance to see if the program lives up to expectations. When it meets with his approval, it is added to the corral, one more animal to help with the ranch work.

Fast-Breeding Ranch, Fast-Breeding Range

Riding a time-sharing program on a batch processing range is like riding a hobbled race horse: one has trouble understanding how it was bred to run.

Riding a batch program on a time-sharing range is like riding a plow-horse: It can pull heavy loads, but slowly and only straight ahead.

One method of promoting the spread of computer ideas is to improve the speed and convenience of breeding programs. This can be done a number of ways.

A most common, but expensive, trick has been for a researcher with a large grant to hire a number of skilled programmers who write and modify programs to produce a string of steeds useful for his own local research.

Another way to improve the convenience of writing programs is through the use of a good time-sharing system. The convenience of a fast-breeding, fast-training, interactive system allows a teacher to handle his own programs. It also means that more of his students know how to program and may be hired part-time. Finally, it means that larger programs (perhaps up to fifty lines) can be locally bred rather than imported from somewhere else through the computer center.

Into the Sunset

The parable makes the obvious prediction that the spread of computing ideas will be aided by:

1. Improving the shipment of the best trained middle sized programs,
2. Increasing the convenience of the local breeding of all programs, and
3. Effectively advertising good ideas to both farmers and ranchers.

But the most important lesson of the parable is that anything that can vary, will vary: the program's behavior, the programmer's ability, the academic's sophistication, the clerk's competence, the climate of research, or the needs of teaching.

Because of this enormous variation, generalizations about the subject die young. A discussion of a computer program is most fruitful when it is most specific. Stories tell more than statistics.

Computer-Based Experiments in Cognitive Psychology

William L. Bewley

Computer Technology Program
Northwest Regional Educational Laboratory
Portland, Oregon 97204

Author is formerly of Dept. of
Psychology at Lawrence University
in Appleton, Wisconsin

EXPERIMENT 1: PATTERN RECOGNITION

This paper describes the first of six computer-based experiments contained in *Cognitive Psychology: A Computer-Oriented Laboratory Manual,* a product of Dartmouth College's project COMPUTe. The six experiments were designed for use in any introductory or advanced undergraduate psychology course either partially or totally concerned with human cognition.

Each experiment is run on a time-sharing system using a terminal as the input/output device. The computer programs which run the experiments require 8K storage and are written in Dartmouth BASIC. Versions which run on DEC'S RSTS system are also available, and the programs are presently being modified to run in HP BASIC.

Each program performs two functions: (1) it runs the student as a subject in an experimental task, e.g., visual search, continuous memory, paired-associate learning, concept learning, a game similar to the Prisoner's Dilemna, and the Missionaries and Cannibals problem; (2) it runs a simulation of an information-processing model on the task, e.g., Pandemonium, the Atkinson and Shiffrin buffer model, Hintzman's discrimination-net model, Levine's focusing model, a modification of Messick's social motives model, and the Newell and Simon General Problem Solver. The student is asked to compare his performance on the task with that of the model and to compare what he thinks he did in the task with what the model says he did. The programs are fairly flexible in that some features of the task and model can be changed by the student so that he can run his own experiments on other students and the model.

Background

According to the Pandemonium model of Selfridge (1959), pattern recognition is a hierarchical process in which information from input patterns is fed through a succession of analyzers or "demons," the output of one demon serving as the input of the next. At the lowest level are image demons which simply form an image or unprocessed copy of the input stimulus. This image is then processed by a set of feature extraction demons which, as their name implies, extract from the image such features as straight lines, curved lines and angles. Cognitive demons then evaluate the extracted features to determine the likelihood that one of the several alternative patterns is the input pattern. These likelihood estimates are then sent to a decision demon which makes the recognition response, deciding that the most likely alternative is the input pattern. There are two strong implications of this model. The first is that there are hierarchical levels of processing in pattern recognition. The second is that the processing at each level is parallel, i.e., all image demons process the input pattern simultaneously, all feature extraction demons process the images simultaneously, and all cognitive demons evaluate the extracted features simultaneously. The purpose of this experiment was to test these two implications.

The Experiment

This experiment is a partial replication of the visual search experiment of Neisser (1963). Each student is shown 20 lists of letters, one list at a time. Each list is composed of 50 lines, 6 letters per line, arranged in 5 rows with 10 lines per row. The student's task is to search each list for a particular target letter, scanning the list from left to right within each row starting with the top row and working down (i.e., the scanning pattern used in reading). A question mark is typed immediately below each list to indicate that the student may begin his search. When he is ready to begin he presses the carriage return button again. The computer then prints the line number at which the target occurred and the student's search time in seconds. The simulation of the Pandemonium model then scans the same list for the same target, and its search time, in arbitrary units, is printed. Following this, the computer types the next list. This procedure is repeated until a search time is obtained for each of the 20 lists.

There are two independent variables: the target to be found and the context in which it is embedded. The five targets are Q, Z, not-Q, not-Z, and Q or Z. If the target is Q, only one of the 50 lines in the list contains a Q. If the target is Z, only one of the 50 lines contains a Z. If the target is not-Q, only one of the 50 lines does not contain a Q. If the target is not-Z, only one of the 50 lines does not contain a Z. If the target is Q or Z, one of the 50 lines contains either a Q or a Z. In every case, the subject is searching for the line containing the target. The context in which the target is embedded is either angular or round. For the angular context, the non-target letters are drawn from the set E, I, M, V, W and X. For the round context, the non-target letters are drawn from the set C, D, G, O, R and U. Since there are five targets and two contexts, there are ten experimental conditions. The 20 lists shown to each subject represent only one experimental condition. That is, the 20 lists have the same target and context. The only difference among the lists presented to each subject is the line at which the target occurs.

The dependent variable is the average time it takes to scan one line. This is calculated by dividing reaction time by the line number at which the target appears.

Predictions

Hierarchical Levels

For a task in which subjects search for a single letter, Neisser (1964) suggests a Pandemonium something like that shown in Figure 1. Only one cognitive demon is present, the cognitive demons for all letters but the target letter being effectively turned off. This implies that the processing of a letter needs to go only as far as the feature extraction level until the target letter is found. If, for example, the search is for a Q, processing of each letter terminates as soon as features not associated with a Q are extracted. The recognition process is completed (i.e., the Q cognitive demon and the decision demon are activated) only when features associated with a Q are extracted. When searching for the absence of a letter (e.g., not-Q), the structure of the Pandemonium is similar to that used in searching for the presence of a letter in that the cognitive demons for all letters but the critical letter (e.g., Q) are turned off. As before, this means that the recognition process is completed only for the critical letter. However, since the critical letter is present in all lines but one, the recognition process must be completed at least once in each line. Since completion of the recognition process (activation of the cognitive and decision demons) requires more time than processing to only the feature extraction level, it should take longer to search for the absence than for the presence of a letter. One prediction derived from the idea of hierarchical levels of processing, then, is that scanning time per line should be greater when the search is for not-Q than for Q, and for not-Z than for Z.

Referring again to Figure 1, it is seen that there are three levels of feature extraction. Level 1 extracts the crudest features, the simple presence or absence of lines, angles and curves. At level 2, the feature extraction is more detailed, producing information regarding the presence or absence of categories of lines (vertical, horizontal and oblique), angles (right and acute), and curves (continuous and discontinuous). Level 3 produces the most detailed information, going beyond the mere presence or absence of the feature categories of level 2 to indicate the number of features in each category. It is assumed that these three levels of feature extraction form a hierarchy, level 1 extraction occurring before level 2, and level 2 extraction occurring before level 3. It is further assumed that the cognitive demon evaluates features in the order of their extraction, level 1 features being evaluated first, then level 2 features, and finally level 3 features. The cognitive demon evaluates the features extracted by level 2 and level 3 only if the features extracted by the preceding level, levels 1 and 2, respectively, indicate that the letter being processed could be the target letter. Since more processing time will be required when more feature extraction levels must be evaluated, and since more feature extraction levels must be evaluated with greater similarity of target-letter and context-letter features, a second prediction derived from the idea of hierarchical levels of processing is that scanning time per line should be greater for a Q-search than for a Z-search in the round context and greater for a Z-search in the angular context.

Parallel Processing

Searching for two target letters, e.g., searching for either a Q or a Z, requires the addition of a cognitive demon to the Pandemonium shown in Figure 1. In general, there will be one cognitive demon for each target letter. Since demons at a particular level of the Pandemonium operate in parallel, however, the addition of a cognitive demon will have no effect on processing time. Accordingly, there should be no difference in scanning time per line in searching for a Q, a Z, or either a Q or a Z. To be more specific, it should take no longer to search for a Q when a Z is also a possible target than to search for a Q as a single target, and vice versa.

USING THE PROGRAM

The program has five running options: (1) to run a subject and the simulation or run the simulation alone; (2) a choice of four targets (Q, Z, Q or Z, or any letter of the alphabet); (3) a goal of detecting the presence of the target in a line or a goal of detecting its absence; (4) a choice of three contexts (round – C, D, G, O, R and U; angular – X, I, M, V, W and X; or any six letters of the alphabet); and (5) to have the Pandemonium search procedure printed or omitted. It is suggested that a student should run himself as a subject in conjunction with the simulation before running his own experiments on other students and the simulation. The choice of target, goal (presence or absence of the target), and context will be determined by the experimental condition under which the program is to be run. For the first run, this should probably be determined for each student by the instructor in order to ensure an approximately equal number of subjects in each experimental condition.

The printout of the Pandemonium search procedure lists the feature extraction levels used in processing each target and context letter. Examining the procedure for each of the experimental conditions should help the student to understand the structure of the Pandemonium and how the experimental effects predicted above are produced by this structure. The search procedure should be printed whenever the student runs a new experimental condition.

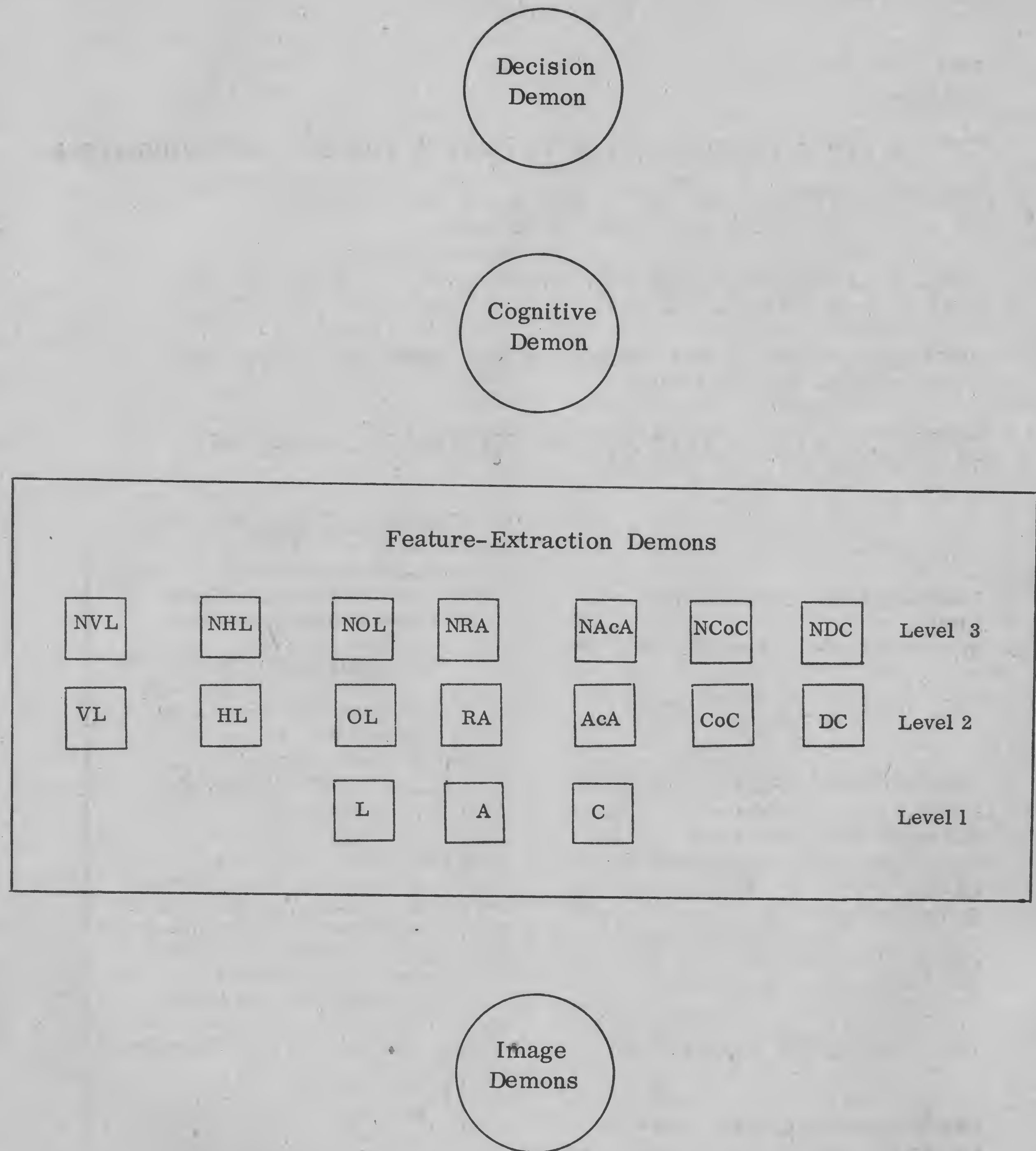


Figure 1. Pandemonium. (L = lines, A = angles, C = curves, V = vertical, H = horizontal, O = oblique, R = right, Ac = acute, Co = continuous, D = discontinuous, N = number.)

A SAMPLE RUN

```
RUN
SNERCH

TYPE 1 FØR A SIMULATIØN, 2 TØ TEST A SUBJECT AND SIMULATE?2

TARGET:  TYPE 1 FØR Q, 2 FØR Z, 3 FØR Q ØR Z,
ØR 4 IF YØU WISH TØ INPUT YØUR ØWN?1

TYPE 1 IF THE GØAL IS THE PRESENCE ØF TARGET LETTER,
2 IF IT IS THE ABSENCE?1

CØNTEXT:  TYPE 1 FØR RØUND, 2 FØR ANGULAR, 3 IF YØU
WISH TØ INPUT YØUR ØWN?1

WØULD YØU LIKE A PRINTØUT ØF THE SEARCH PRØCEDURE?
TYPE 1 FØR YES, 0 FØR NØ?1


PANDEMØNIUM SEES:  'C'
LEVEL 1
ELIMINATES:  D G R U Q
POSSIBILITIES REMAINING:  C O
THE CHARACTER IS CØNTEXT
```

Pandemonium's processing of the letter 'C'
Feature-extraction level 1 eliminates all letters having lines or angles (the 'C' has only a curve)
Since the target letter, 'Q,' is among those eliminated, Pandemonium can conclude that the letter is context and go on to the next letter

```
PANDEMØNIUM SEES:  'D'
LEVEL 1
ELIMINATES:  C Ø U
POSSIBILITIES REMAINING:  D G R Q
LEVEL 2
ELIMINATES:  R
POSSIBILITIES REMAINING:  D G Q
LEVEL 3
ELIMINATES:  G Q
POSSIBILITIES REMAINING:  D
THE CHARACTER IS CØNTEXT
```

Pandemonium's processing of the letter 'D'

Feature-extraction level 1 eliminates all letters not having lines, angles, and curves (the 'D' has all three). The program treats the 'O' as being without a slash.
Feature-extraction level 2 eliminates all letters having oblique lines, acute angles, and continuous curves (a 'D' has none of these)
Feature-extraction level 3 eliminates all letters not having one vertical line, two horizontal lines, two right angles and one discontinuous curve (the 'D' has all of these)

```
PANDEMØNIUM SEES:  'G'
LEVEL 1
ELIMINATES:  C Ø U
POSSIBILITIES REMAINING:  D G R Q
LEVEL 2
ELIMINATES:  R
POSSIBILITIES REMAINING:  D G Q
LEVEL 3
ELIMINATES:  D Q
POSSIBILITIES REMAINING:  G
THE CHARACTER IS CØNTEXT
```

```
PANDEMØNIUM SEES:  'Ø'
LEVEL 1
ELIMINATES:  D G R U Q
POSSIBILITIES REMAINING:  C O
THE CHARACTER IS CØNTEXT

PANDEMØNIUM SEES:  'R'
LEVEL 1
ELIMINATES:  C Ø U
POSSIBILITIES REMAINING:  D G R Q
LEVEL 2
ELIMINATES:  D G Q
POSSIBILITIES REMAINING:  R
THE CHARACTER IS CØNTEXT

PANDEMØNIUM SEES:  'U'
LEVEL 1
ELIMINATES:  C D G Ø R Q
POSSIBILITIES REMAINING:  U
THE CHARACTER IS CØNTEXT

PANDEMØNIUM SEES:  'Q'
LEVEL 1
ELIMINATES:  C Ø U
POSSIBILITIES REMAINING:  D G R Q
LEVEL 2
ELIMINATES:  R
POSSIBILITIES REMAINING:  D G Q
LEVEL 3
ELIMINATES:  D G
POSSIBILITIES REMAINING:  Q
THE CHARACTER IS A TARGET


PRØBLEM 1


ØRCGDU RGCØUD UDRCGØ CRØGUD CØRUDG ØURCDG ØGCRUD RØUGDC CØRGUD CDRGØU
UØCDGR DGCRUØ ØRUCGD UCRGØD RØDCGU UØDRCG UDCGØR GURCØD ØURCDG RGØUDC
DUØGRC ØDUCRG UDGCRØ DCUØGR DCRUGØ GDCØRU ØRCUGD ØRDUCG RCØDUG ØGUCDR
DGCRUØ CRDUØG ØDUCRG URDCGØ DRUCGØ CØGURD RGCUDØ UCDØGR GØCDUR DØRUCG
CRØUGD ØRCUDG DUGRCØ GUDCØR CØDUGQ GUØDCR CRØUGD ØRUCGD DØCGUR RCGØUD

?
?
```

LINE 45   SUBJECT TIME 30 ←——— in seconds

LINE 45   SIMULATED TIME 4960 ←——— arbitrary time units

And so on for 19 more problems.

PROGRAM LISTING

```
0 REM NEISSER'S VISUAL SEARCH AND PANDEMØNIUM
2
4 RANDØMIZE
6
8 REM D$ CØNTAINS THE LETTERS A-Z
10 DIM D$(26)
12 REM B HØLDS 3 FEATURE-EXTRACTIØN LEVELS FØR 26 LETTERS
14 DIM B(26,3)
16 REM FØR EACH ØF 20 TRIALS, Ø TELLS WHICH ØF THE 50 SETS
18 REM ØF SIX CHARACTERS HØLDS THE ANØMALY
22 DIM L$(5),L(5),R(5)
24 REM A$ AND R$ HØLD THE ANGULAR AND THE RØUND CØNTEXT LETTERS
26 DIM A$(6),R$(6)
28 REM C$ AND C HØLD THE CØNTEXT CHARACTERS ACTUALLY USED
30 DIM C$(6),C(6)
32 REM E$ AND E HØLD A SET ØF SIX LETTERS TØ BE PRINTED
34 DIM E$(6),E(6)
36 DIM T$(11),T(11),U(11)
38 REM V(I) IS THE NUMBER ØF LEVELS TAKEN TØ RECØGNIZE THE I'TH LETTER
40 DIM V(26)
42
44 REM FØR THE L, C, AND E ARRAYS, THE STRING ARRAY HØLDS THE
46 REM ACTUAL LETTER, WHILE THE NUMERIC ARRAY HØLDS ITS
48 REM PØSITIØN IN THE ALPHABET
50
60 REM FNT FINDS THE TIME ØF DAY IN SECØNDS.  NØTE, THIS
62 REM FUNCTIØN WILL PRØBABLY VARY FRØM CØMPUTER TØ CØMPUTER.
64 REM IN MANY VERSIØNS ØF BASIC, IT MAY BE DEFINED AS:
66 REM          DEF FNT=TIME(0)
68 DEF FNA(X)=INT(X*RND+1)
70 DEF FNT
72 LET C$=CLK$
74 LET FNT=3600*VAL(SEG$(C$,1,2))
76 LET FNT=FNT+60*VAL(SEG$(C$,4,5))+VAL(SEG$(C$,7,8))
78 FNEND
80
82
100 FØR M=1 TØ 26
110 READ D$(M)
120 NEXT M
130 FØR M=1 TØ 26
140 FØR M1=1 TØ 3
150 READ B(M,M1)
160 NEXT M1
170 NEXT M
180 FØR M=1 TØ 6
190 READ A$(M)
200 NEXT M
210 FØR M=1 TØ 6
220 READ R$(M)
230 NEXT M
240 PRINT "TYPE 1 FØR A SIMULATIØN, 2 TØ TEST A SUBJECT AND SIMULATE";
250 INPUT Z4
260 IF Z4=1 THEN 340
270 LET Z4=0
280
290 REM WHEN THE FØLLØWING RØUTINE IS FINISHED, T9 WILL CØNTAIN
300 REM THE NUMBER ØF TARGET LETTERS, THE L$ ARRAY WILL CØNTAIN THE
310 REM LETTERS, AND THE L ARRAY WILL CØNTAIN THE LETTERS'
320 REM ALPHABET PØSITIØNS
330
340 PRINT
350 PRINT "TARGET:  TYPE 1 FØR Q, 2 FØR Z, 3 FØR Q ØR Z,"
360 PRINT "ØR 4 IF YØU WISH TØ INPUT YØUR ØWN";
370 INPUT T
380 IF T>4 THEN 350
390 ØN T GØTØ 550,590,630,400
400 PRINT
410 PRINT "HØW MANY DIFFERENT TARGET LETTERS";
420 INPUT T9
430 PRINT "TYPE THEM ØNE TØ A LINE"
440 FØR T1=1 TØ T9
450 INPUT L$(T1)
460 FØR R=1 TØ 26
470 IF D$(R)=L$(T1) THEN 520
480 NEXT R
490 PRINT "SØRRY, '";L$(T1);"' IS NØT A LETTER."
500 PRINT "PLEASE TYPE IT ØVER";
510 GØTØ 450
520 LET L(T1)=R
530 NEXT T1
540 GØTØ 700
550 LET T9=1
560 LET L$(T9)="Q"
570 LET L(T9)=17
580 GØTØ 700
590 LET T9=1
600 LET L$(T9)="Z"
610 LET L(T9)=26
620 GØTØ 700
630 LET T9=2
640 LET L$(1)="Q"
650 LET L(1)=17
660 LET L$(2)="Z"
670 LET L(2)=26
680
690
700 PRINT
710 PRINT "TYPE 1 IF THE GØAL IS THE PRESENCE ØF TARGET LETTER,"
720 PRINT "2 IF IT IS THE ABSENCE";
730 INPUT T7
740 IF T7<>2 THE 790
750 PRINT
760 PRINT "TYPE 1 IF PANDEMØNIUM IS TØ STØP SCANNING A LINE UPØN"
770 PRINT "REACHING A TARGET, 2 IF IT SHØULD SCAN THE ENTIRE LINE"

780 INPUT Z6
790 PRINT
800 PRINT "CØNTEXT:  TYPE 1 FØR RØUND, 2 FØR ANGULAR, 3 IF YØU"
810 PRINT "WISH TØ INPUT YØUR ØWN";
820 INPUT C
830 IF C>3 THEN 800
840 ØN C GØTØ 850,890,930
850 FØR M=1 TØ 6
860 LET C$(M)=R$(M)
870 NEXT M
880 GØTØ 950
890 FØR M=1 TØ 6
900 LET C$(M)=A$(M)
910 NEXT M
920 GØTØ 950
930 PRINT "TYPE THE SIX LETTERS, SEPARATED BY CØMMAS";
940 INPUT C$(1),C$(2),C$(3),C$(4),C$(5),C$(6)
950 FØR M=1 TØ 6
960 FØR M1=1 TØ 26
970 IF C$(M)=D$(M1) THEN 1010
980 NEXT M1
990 PRINT "SØRRY, '";C$(M);"' IS NØT A LETTER."
1000 GØTØ 930
1010 LET C(M)=M1
1020 NEXT M
1030
1040 REM SET UP THE T ARRAY TØ CØNTAIN THE CØNTEXT
1050 REM LETTERS AND THE TARGET LETTERS \
1060
1070 FØR M=1 TØ 6
1080 LET T(M)=C(M)
1090 LET T$(M)=C$(M)
1100 NEXT M
1110 REM T1 IS THE LENGTH ØF THE T ARRAY
1120 LET T1=6+T9
1130 FØR M=7 TØ T1
1140 LET T$(M)=L$(M-6)
1150 LET T(M)=L(M-6)
1160 NEXT M
1170 PRINT
1180 PRINT "WØULD YØU LIKE A PRINTØUT ØF THE SEARCH PRØCEDURE?"
1190 PRINT "TYPE 1 FØR YES, 0 FØR NØ";
1200 INPUT Z
1210 IF Z=0 THEN 1240
1220 PRINT
1230 PRINT
1240 FØR M=1 TØ T1
1250 IF Z=0 THEN 1290
1260 PRINT
1270 PRINT
1280 PRINT "PANDEMØNIUM SEES:  '";T$(M);"'"
1290 FØR I=1 TØ T1
1300 LET U(I)=0
1310 NEXT I
1320 FØR M1=1 TØ 3
1330 IF Z=0 THEN 1360
1340 PRINT "LEVEL";M1
1350 PRINT "ELIMINATES: ";
1360 LET B=B(T(M),M1)
1370 REM B IS LEVEL M1 FØR THE LETTER T$(M
1380 REM NØW FIND WHICH ØF THE T1 LETTERS THIS LEVEL ELIMINATES
1390 FØR M2=1 TØ T1
1400 IF U(M2)=1 THEN 1450
1410 IF B(T(M2),M1)=B THEN 1450
1420 IF Z=0 THEN 1440
1430 PRINT " ";T$(M2);
1440 LET U(M2)=1
1450 NEXT M2
1460
1470 LET P1=P2=0
1480 IF Z=0 THEN 1510
1490 PRINT
1500 PRINT "PØSSIBILITIES REMAINING: ";
1510 FØR M2=1 TØ T1
1520 IF U(M2)=1 THEN 1610
1530 IF Z=0 THEN 1550
1540 PRINT " ";T$(M2);
1550 IF M2<=6 THEN 1590
1560 LET P2=1
1570 REM THE LETTER MIGHT STILL BE A TARGET
1580 GØTØ 1610
1590 LET P1=1
1600 REM THE LETTER MIGHT STILL BE CØNTEXT
1610 NEXT M2
1620 IF Z=0 THEN 1640
1630 PRINT
1640 IF P1+P2=2 THEN 1710
1650 IF Z=0 THEN 1720
1660 IF P2=1 THEN 1690
1670 PRINT "THE CHARACTER IS CØNTEXT"
1680 GØTØ 1720
1690 PRINT "THE CHARACTER IS A TARGET"
1700 GØTØ 1720
1710 NEXT M1
1720 LET V(T(M))=M1
1730 NEXT M
1740
1750
1760 FØR M=1 TØ 20
1770 IF M>6 THEN 1800
1780 LET A=FNA(6)
1790 GØTØ 1810
1800 LET A=6+FNA(14)
1810 IF Ø(A)<>0 THEN 1770
1820 READ Ø(A)
1830 NEXT M
1840 FØR I=1 TØ 20
1850 PRINT
1860 PRINT
1870 PRINT
1880 PRINT "PRØBLEM";I
1890 PRINT
```

```
1900 PRINT
1910 LET V=0
1920 FØR J=1 TØ 50
1930 FØR M=1 TØ 6
1940 LET E(M)=0
1950 NEXT M
1960
1970 REM PUT THE CØNTEXT LETTERS IN RANDØM ØRDER IN THE E ARRAYS
1980 FØR M=1 TØ 6
1990 LET A=FNA(6)
2000 IF E(A)<>0 THEN 1990
2010 LET E(A)=C(M)
2020 LET E$(A)=C$(M)
2030 NEXT M
2040 REM IN CASE THERE IS NØ TARGET LETTER IN THIS SET ØF SIX,
2050 REM PANDEMØNIUM MUST CHECK ALL SIX.
2060 LET A=6
2070
2080 REM NØW STICK IN A TARGET IF APPRØPRIATE
2090 IF T7=1 THEN 2120
2100 IF Ø(I)=J THEN 2170
2110 GØTØ 2130
2120 IF Ø(I)<>J THEN 2170
2130 LET A=FNA(6)
2140 LET A1=FNA(T9)
2150 LET E$(A)=L$(A1)
2160 LET E(A)=L(A1)
2170 IF Z4=1 THEN 2240
2180 FØR M=1 TØ 6
2190 PRINT E$(M);
2200 NEXT M
2210 PRINT " ";
2220 IF J/10 <> INT(J/10) THEN 2240
2230 PRINT
2240 IF J>Ø(I) THEN 2330
2250 IF Z6<>2 THEN 2300
2260 LET A=6
2270
2280 REM ADD ØN THE TIMES FØR ALL THE LETTERS UP TØ AND
2290 REM INCLUDING THE TARGET LETTER
2300 FØR M=1 TØ A
2310 LET V=V+V(E(M))
2320 NEXT M
2330 NEXT J
2340 IF Z4=1 THEN 2430
2350 PRINT
2360 REM NØW TYPE A QUESTIØN MARK AND WAIT FØR THE USER TØ PUSH 'RETURN'
2370  INPUT Z9$
2380 LET T0=FNT
2390  INPUT Z9$
2400 LET T0=FNT-T0
2410 PRINT
2420 PRINT "LINE";Ø(I);"SUBJECT TIME";T0
2430 PRINT
2440 PRINT "LINE";Ø(I);"SIMULATED TIME";V*10
2450 NEXT I
2460
2470 DATA "A","B","C","D","E","F","G","H","I","J","K","L","M","N","Ø"
2480 DATA "P","Q","R","S","T","U","V","W","X","Y","Z"
2490
2500 DATA 3,23,138,      7,35,220,      4,18,53,      7,35,135
2510 DATA 3,17,132,      3,17,102,      7,35,105,     3,17,117
2520 DATA 3,17,122,      5,21,61,       3,29,134,     3,17,52
2530 DATA 3,21,136,      3,21,96,       4,16,46,      7,35,155
2540 DATA 7,35,130,      7,41,176,      4,18,88,      3,17,72
2550 DATA 5,21,66,       3,19,74,       3,19,154,     3,19,99
2560 DATA 3,21,81,       3,23,108
2570
2580 DATA "E","I","M","V","W","X"
2590 DATA "C","D","G","Ø","R","U"
2600 DATA 5,6,25,30,45,46,9,11,14,16,19,21,24,26,29,31,34,36,39,41
2610
2620 END
```

REFERENCES

Neisser, U. "Decision-Time without reaction-time: Experiments in visual scanning." *American Journal of Psychology,* 1963, 76, 376-385.

Neisser, U. "Visual search." *Scientific American,* 1964, 210, 94-102.

Selfridge, O. "Pandemonium: A paradigm for learning." *Symposium on the Mechanization of Thought Processes.* London: HM Stationery Office, 1959.

Did You Ever Talk With A Computer?

AIP/360 – PLEASE SIGN ON: 16

TERMINAL N SIGNED ON, 02/15/71

UNFORMATTED MODE

AIP/360 – PLEASE SIGN ON: 16

TERMINAL N SIGNED ON, 02/15/71

UNFORMATTED MODE
clear

CLEARED ONE MESSAGE WAITING

SIGNAL WHEN READY
ready

WOULD YOU LIKE TO STUDY SCIENCE?
I don't know.

THAT IS A GOOD ANSWER. THINK AGAIN.
WOULD YOU LIKE TO STUDY SCIENCE?
what for?

YOU HAVE GIVEN IT A LITTLE THOUGHT.
SOCIETY NEEDS PEOPLE LIKE YOU.
what for?

SOCIETY NEEDS TO MAKE BETTER USE OF ITS
NEW TECHNOLOGY. DO YOU LIKE PHYSICS?
I don't know – what's physics?

PHYSICS IS CONCERNED WITH THE PROPERTIES OF MATTER
AND THE LAWS WHICH DESCRIBE ITS BEHAVIOR.
what's that got to do with me?

ILLEGAL ENTRY. LINE CANCELLED. TRY
ANOTHER RESPONSE.
why should I know about science?

SCIENCE IS INTERESTING. IT HELPS US TO
UNDERSTAND THE WORLD IN WHICH WE LIVE.

Squaresville

Arrange two of each of the digits 0 to 9 so as to form a 20-digit number. Your number may not begin with a zero. Then score your number as follows:

For every two consecutive digits that form a perfect square, score two points. For every three consecutive digits that form a perfect square, score three points. A four-digit square scores four points, and so on.

For example, if your number was 58738219024971950364, you would score two points for 49, two points for 36, two points for 64, and six points for 219024 – for a total of 12 points. You may *not* count 036 as a three-digit square.

What is the maximum number of points you can score?

CREATIVE COMPUTING
Feature Review

George Leonard's View of the Computer in Education

Alfred Bork
Physics Computer Development Project
University of California
Irvine, California 92664

Workers developing educational materials using the computer can, and often do, simply start where they are, with whatever equipment they can acquire, and chip away at courses.

However, effective long-range plans must be much more systematic and detailed. We can often distinguish readily between short-range goals, what we are immediately trying to do with projects, and long-range goals. Long-range goals can be formulated only if we decide how the future *should* be shaped. Individuals with strong views about the future, with powerful ideas about what role computers should play in education twenty or twenty-five years from now, not only offer guidance, but may well influence the future; a prophecy of this kind can be a self-fulfilling prophecy.

Relatively few have attempted to view the future of computers in education. Developers who continue working where they are, with the equipment they have on hand, often become fierce defenders of that equipment, arguing its merits, rather than, perhaps more rationally, striving to improve it.

Science fiction has offered some views of a computerized future that mention educational aspects, in such works as Arthur Clark's *The City and the Stars*. Often the views presented in such literature are hostile. For example, the computer-teacher in Zamatien's *We* is not a positive thing, but aids in dehumanizing the society.

But George Leonard's book, *Education and Ecstasy*, shows a positive, and, I believe, extremely interesting view of the role computers may play in learning. This view is almost unknown to the developers of learning materials employing the computer, so I intend to review Leonard's picture of the future. This, like all predictions, must be taken with a grain of salt; Leonard suggests as much in the introduction to the two chapters of the book that will concern us. One of my knowledgeable friends has characterized the passages as romantic. I believe, nevertheless, that they do represent an interesting view of the future, even if time should prove them not accurate in *all* details.

Leonard's book combines two seemingly unrelated trends in modern society, the encounter group philosophy of such establishments as Esselin, and the technological, even Skinnerian, approaches to education, particularly those involving the computer. In earlier didactic chapters he brings these possibilities together while criticizing education as it exists today.

The main interest in *Education and Ecstasy*, to me, is in the chapters that portray a school of the year 2001. The description uses the fictional guise of a visit by parents to the school to watch the progress of their children. We get a panoramic view of the learning activity, and we see the overall "philosophy" underlying the school.

The learning structure pictured is completely dependent on the existence of the computer, and would be impossible without advanced technology. The computer technology depicted is graphic, employing large three-dimensional, color pictures, and sound, in addition to alphanumeric interactions.

The main arena for the knowledge-based forms of education is the Basics Dome. Students enter and leave freely; the entire school is unscheduled, stressing that students do not appear for "classes" at any particular time inasmuch as there *are* no classes!

The students are three to ten year olds. After age ten people are expected to know all the "basic" information, including calculus! Since the time is free, enormous thought has to be given to motivational issues, so that the students will *want* to do the necessary tasks, rather than be coerced to do them as is often the stiuation in schools today.

The educational view presented, the free learning view is a natural extension of the self-paced or Keller plan ideas that are now coming into practice, with much more emphasis given to complete freedom from scheduling, and with the material much more highly individualized than at present.

Students entering the Basics Dome see a circular ring of computer consoles around the outer wall. Each has a keyboard, allowing access to *all* human symbols, not just the restricted or full ASCII, or APL set, typical today. The technology for allowing access to all symbols is available and has been built into the PLATO system; but the standard terminals today deny us this feature. Each student has headphones for audio messages. The display is a large three-dimensional holographic display in full color. The displays touch each other at the edges, so that the room has a continuous band of pictorial information. This touching is more than simply physical; the computer is clever enough to have the displays interact, and information may spread over more than one display, moving and contracting. Furthermore, displays will have in their intermediate areas related aspects of what students at each of the stations are doing.

Students are identified to the computer by means of an electronic identification device, which they attach to the chair. This device also does queueing. The computer has complete records of students' efforts, so whenever students resume work it is all ready. A session starts with review and then moves on to new items. The individual sessions shown are not long.

A sample session of language learning is sketched. As one would expect, this is difficult to do, and, while interesting, is perhaps one of the weaker features of the chapter. Writers about the future often have this problem. It is easier to imagine the overall structure than to delineate concrete details. This sketch is done much better in this book than in B. F. Skinner's *Walden II*, where a great discrepancy separates the glowing philosophy and the mundane details.

Graphics play a vital role. This trend is already established in contemporary education, but still is not as widely recognized as it should be. Almost all of the current major educational developmental projects in the United States are graphic based. I regard it as an historical accident that earlier terminals were nongraphic, forcing users of the computer in learning situations to begin with nonpictorial formats. The role of pictures is so important in all educational processes, in a variety of levels, that this seems to me to be an intolerable situation. Now that reasonably priced and highly reliable graphic terminals are available, I expect the situation to change. We could even argue that extensive nongraphic developmental work is a waste of time today. The future of computers in education will almost certainly be highly pictorial, allowing teachers to access to these important nonverbal learning techniques.

The learning environment in *Education and Ecstasy* is a computer-managed environment, with the computer knowing the educational progress of each student and making judgments based on this knowledge within the learning material. Each student is an individual with a highly individualized learning sequence. This is a long-standing goal for computers in education, even though it remains difficult to achieve in full-scale systems. We are now beginning to see systems that do this, systems where extensive memory of students' efforts and achievement both on and off the computer, is accessible to the program.

At least two aspects of the technology pictured are far beyond anything possible today (and probably beyond what will be available in 2001). The first, in active use in the school we visit, is the use of brain wave information within the learning dialogs. The headset which brings the sound also picks up brain patterns, allowing the computer to determine what the student is absorbing, whether the material needs to be reviewed, whether the program can accelerate.

The second advanced technological innovation, more radical and "criticized" by the "conservative" school director, involves direct brain manipulation, bypassing the senses entirely. Naturally the details are vague and it is not clear if the criticism is tongue-in-cheek or is the usual resistance to new educational developments. Contemporary readers are almost certain to approach this with apprehension; such future possibilities have often been the subject of frightening fictional presentations, such as John Hershey's *The Child Buyer*. It seems unlikely that any such technique would be in use in just twenty-five years.

I hope I have encouraged at least some of you to read the book, which includes many more details.

How realistic is this view of the future? What alternate patterns are plausible? In discussions with friends, alternative views have been expressed, both the view that the picture presented is much too radical a change to occur in twenty-five years, and the view that changes will be much more drastic than those suggested. I don't know how to pick between these two positions! The notion that educational change occurs only slowly is ingrained, and does seem to have empirical basis. The vested interests in maintaining the system as it is are powerful. So desirable educational change is not rapid.

On the other hand, we are in a period of very rapid change in computer technology. Computers are becoming more economical while everything else is increasing in cost, promising that highly computerized educational systems *will* come into widespread use in the 80's and 90's. Economic considerations alone will be an important factor in pressing for such change, provided viable teaching materials can be developed in sufficient time.

This last factor, the existence of the pedagogical and programming skills discussed in *Education and Ecstasy*, is much more questionable than the hardware. None of our current computer-based education projects show such sophistication in computer use in educational situations, although many projects are striving to use computers more effectively. We still have a long way to go.

Even the hardware aspect is not entirely clear. The environment projected is a timesharing environment, with the central computer holding the record capabilities. There is probably local processing at the displays. Except for record keeping and large databases, future use may not be in the timesharing mode, but may tend toward sophisticated stand-alone machines. A striking aspect of recent technology has been the development of more and more compact computers. We have now many competing "computer-on-a-chip" assemblies which can be put together to form systems. This development of microcomputer technology will

CREATIVE COMPUTING Reviews

(con't) George Leonard's View of the Computer in Education

continue, with units becoming cheaper, faster, and easier to assemble for the purposes at hand.

To think of today's minis is misleading. The power of such systems in the near future will be more comparable to that of very large contemporary computers, even though these systems will be largely self-contained and stand-alone. Modern video-based technology will, I believe, have great ramifications too; it also suggests the possibility of very powerful local processing. The local processor can drive displays without timesharing limitations, and interactive computer graphics can overcome the limitations of a 1200 baud connection.

George Leonard's view of the future is only one of many. But such speculation, such description of ideal future conditions in a broad and sweeping sense, is important for developers of computer-based material. It is easy to become frozen in the hardware and technology available at the moment, and so waste years of time preparing materials that will only be obsolete when they are finished. Perhaps nowhere is the future shock phenomenon likely to be more important than in areas touched by the computer, because of the very rapid advances. Whether you accept George Leonard's view or not, or perhaps just deem it an interesting possibility, the need for long-range thinking and speculation about the future is important for all of us.

THE COMPUTER: HOW IT'S CHANGING OUR LIVES, Joseph Newman (Editor), U. S. News and World Report Publishing, Washington, DC 20037, 1972.

An "unscary" straight forward book about what computers can do and cannot do is how I would summarize THE COMPUTER. The book was obviously written for the general non-computer knowledgeable public with the intent to explain how computers affect the everyday man. I believe it achieves this objective quite well. It could additionally serve as a classroom source on computers for a social science, history or science class.

The book seems to be written at a level between high school and college and might prove difficult for some students. I found the writing style rather interesting: It seemed at times that I was reading a script for a Walter Cronkite CBS Special Report program. The book seeks to present both the benefits and dangers of computer involvement in the fields of medicine, education, science, communications and government. In addition it deals frankly with the very current issue of privacy and also takes a look at "computer" mistakes.

I would recommend the book for the reader who is curious or even alarmed at how the computer is "changing our lives". It offers the reader an easily understandable, thorough discussion without the necessity of learning "computerese"

Daniel A. Myers
Colorado Springs, CO

MAN AND THE COMPUTER, John G. Kemeny, 151pp. $6.95, Charles Scribner's Sons, New York. 1972.

Although this book is based on a series of lectures delivered three years ago, it deserves a careful reading. It is written for the person unfamiliar with computers but could easily be read by computer professionals who need a new perspective of the computer. The author divides the book into two parts. The first part is a short history of computers and time-sharing plus speculation as to whether we should consider the computer a new species. The second part is a peek into 1990 to see how this relation between man and computer might develop.

From the development of stored programs with the leadership of John von Neumann, the computer exhibits characteristics of a species; i.e., metabolism, locomotion, reproduction, individuality, intelligence, and naturalness. However, this flies in the face of current acceptance of the definition of a "live species." The author, after a very clear explanation of Dartmouth Time Sharing, shows how the species of man and machine are coexisting. He then raises the question about this relationship – Is it symbiotic or parasitic? Humans fear that which they do not understand, and the computer professionals have not done a very good job of dispelling this fear. Symbiote or parasite – it is up to man to use "enough understanding and enough foresight" to "assure that the interaction between the two species will be totally beneficial to mankind."

1990 augurs widespread computer usage in business, education, and every day existence. With specific examples the author relates how the technology of 1971 could develop these relationships. Progress in computers has bounded forward with reduced costs in the use of micro-computers, computer terminals, and inter-computer communication. Project INTREX has demonstrated the possibility of storing vast amounts of information, but not economically yet. Hewlett-Packard has produced a personal programmable computer for under $800. Therefore, we humans need the help of a federally subsidized private agency to develop computer systems for public use. This agency could lead the way in controlling this "symbiotic evolution" so that we could experience a "new golden age for mankind."

Edgar T. Canty
Boston, MA.

Computers and Society. Stanley Rothmann and Charles Mosmann. 337 pp. Science Research Associates, Inc.

This textbook is designed for use in a one-term introductory course on computers and their impact on society. Considering the difficulties one faces in trying to decide what such a book should include, and the level at which it should be presented, the authors deserve considerable credit for doing as good a job as they have. But if I had to teach a course on computers and society, there are other books that I would be happier to teach it from. I would, I think, be hard to find the right-kind of person to teach a course based on this book. The book is not technical enough for a technologist and not humane enough for a humanist.

The authors are clearly more at home with technical matters than they are with social implications. Part II, which explains what computers are, is very well done indeed. In the span of not much more than a hundred pages it explains a great deal about computers and explains it well. A student could find plenty to chew on here. Indeed, he may find too much.

But once the authors turn to the social implications of computers, the material gets pretty soggy. Part I is short and is intended, the Instructor's Guide tells us, to stimulate students. But one wonders what sort of student would be stimulated by such statements as "Morality may be viewed as a body of rules defining the individual's relationship to the social group," or "Can we have this freedom within our current system of government and law? Yes! The mechanisms to control technology and its employers are there."

Part III deals with the applications of computers and with the social implications of these applications. Part IV asks how we might control the use of computers and Part V speculates about the future. The book goes downhill as it gets less technical and focusses more on social implications. By the end of the course, the students are assumed to have scaled such heights of mediocrity that they are capable of devoting themselves to exercises like these: "How do you feel about the relative importance of work and leisure in your future life?" (Exercise 6, page 232) "Select a press release implicating the computer in a social mishap and analyze it for sensationalism or biased reporting." (Exercise 5, page 272) "Write an essay giving your opinion about whether research on computer learning should be pursued?" (Exercise 3, page 319). The teacher who likes to assign such exercises would probably like this book.

The job that the authors have tried to do is well worth doing and there will be some who find that the way they have done it in this book suits their tastes. However, I find this book too bland when it comes to social matters and possibly a bit too hard (considering the intended audience) when it comes to technical matters. The technical parts, though good, might be tough sledding because they cover so much in so few words. On the other hand, the non-technical parts are thin and seem intended to give the student the warm feeling of learning something or other without ever having worked very hard to learn it.

It may be that the job the authors have tried to do cannot be done to everyone's satisfaction. One cannot help thinking that mixing computer science and social science is a bit like mixing dill pickles with hot fudge sauce. Though the ingredients are tasty taken separately, they are not easy to mix well into a single course. The authors have tried to bring their ingredients together by eliminating some of the strong flavors of each. While it is true that they have done better in presenting the pickles of computer science than the hot fudge of social science, the mixture does not come off well as a single course. One feels the lack of a unifying concept. One cannot help but ask "Just what is the student of this textbook supposed to learn?"

Excluding the artwork, which has all the charm of a Bulgarian textbook on embalming, the book is attractively put together. it is amply illustrated and, considering how hard it is to illustrate any book about computers, the illustrations are very apt. The cartoons and illustrations drawn from advertisements more than make up for the standard but rather dull, shots of equipment and installations.

The authors claim to have used this book successfully at the college level but it is written clearly enough to be usable in high schools too. The teacher who can provide the unifying concept that the book seems to lack, who can lead his students through the excellent, but difficult, technological parts and who can beef up the rather weak material on social implications will find many rewarding things in this text.

Peter Kugel
Boston, MA.

Getting Started in Classroom Computing. David H. Ahl. 29 pp. $1.50. Digital Equipment Corp., Maynard, Mass. 01754. 1974.

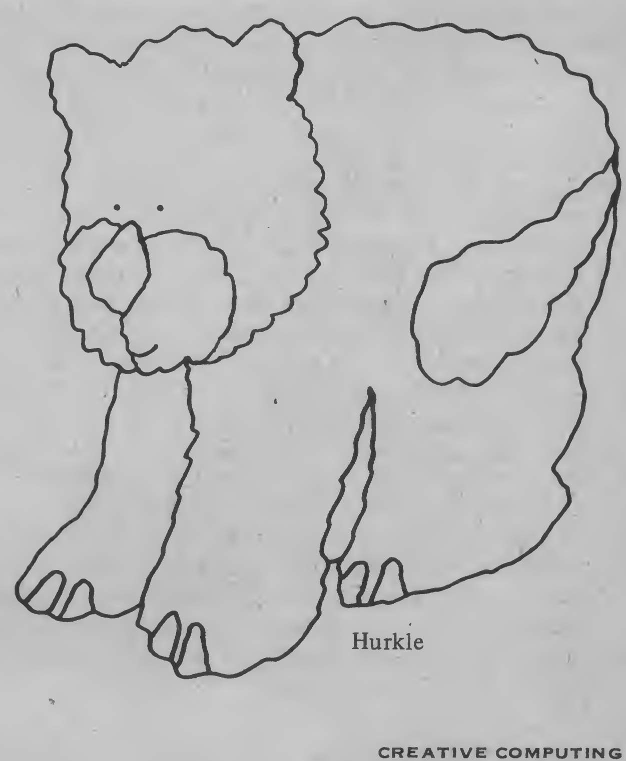
The six games in this booklet are to introduce the newcomer to using games and computers in the classroom. You don't need a computer to play the games but it's more fun. The games are described in the contents as:

- Secret Codes – Introduction to the way punched cards and tapes work
- Guess – Discover an efficient method of searching for a mystery number
- Hurkle – An introduction to grids and coordinate systems
- Bagels – An introduction to mathematical logic
- Caves – Learn to creatively compare similarities and differences of objects

Each game has clear rules, a sample computer run, and ways to be used in the classroom.

I've had 7th and 8th graders play GUESS, HURKLE, and BAGELS – BAGELS is their special favorite and mine. I feel the games could easily be played and enjoyed by elementary students too. Easily read in one sitting, I recommend this booklet for the newcomer to games or computers. For the more advanced, I would suggest the companion books: *101 BASIC Computer Games* and *Understanding Mathematics and Logic Using BASIC Computer Games* by the same author.

Jim Albright
Springfield, OR



Hurkle

creative computing

WHAT IS CREATIVE COMPUTING?

CREATIVE COMPUTING is a lively new magazine for students and teachers using computers in colleges, junior colleges, secondary schools, and even the lower grades. It contains games, simulations, problems, exercises, curriculum materials, and ideas directly usable in the classroom.

CREATIVE COMPUTING deals with the use of computers and computer related devices in mathematics, science, social science, ecology, computer familiarization, computer science, and career education. The content of CREATIVE COMPUTING reflects the view that computers can make learning fun!

CREATIVE COMPUTING provides evaluative reviews of computer hardware, software, applications material, learning aids, books, games, and related devices. It reports on successful experiences of educational users and provides a vehicle for the interchange of programs and materials. CREATIVE COMPUTING also brings its readers capsule summaries of significant educational and computer conferences and meetings.

CREATIVE COMPUTING is a forum for the discussion of the social impact of the computer with a focus on privacy, automation and jobs, leisure time, medical care, pollution control and the like.

CREATIVE COMPUTING is published bi-monthly; each issue containing between 48 and 72 pages of editorial material. The primary objective of CREATIVE COMPUTING is to bring high quality, useful information to students and educators at a reasonable cost. Try it for a year! You'll like it!

CREATIVE COMPUTING BRINGS TOGETHER LEADING PROJECTS AND WRITERS UNDER ONE COVER

CREATIVE COMPUTING is a happy, innovative new magazine! One of its roles is to act as a vehicle for the interchange of information from well-known and not-so-well-known computer education projects. Here are some of the many projects that have agreed to work with CREATIVE COMPUTING!

SECONDARY EDUCATION

Project DELTA
University of Delaware

Huntington Two Project
SUNY, Stony Brook

Project LOCAL
Westwood, MA.

Project SOLO
University of Pittsburgh

TIES
St. Paul, MN.

ALL EDUCATIONAL LEVELS

PLANIT Project
NW Regional Educational Lab

PLATO Project
University of Illinois

TICCIT Project
Brigham Young University

ELEMENTARY EDUCATION

Project LOGO
Mass. Institute of Technology

HIGHER EDUCATION

Project CBE
University of Texas

Project COMPUTe
Dartmouth College

Project CONDUIT
Univ. of Iowa (and others)

Project EXTEND
University of Michigan

Physics Computer Development Project
University of California—Irvine

CREATIVE COMPUTING SUBSCRIPTION FORM

* a non-profit magazine of educational and recreational computing *

* published bi-monthly -- 6 issues per year *

INSTITUTIONAL
- ☐ 1-year $15
- ☐ 3-year $40

INDIVIDUAL REGULAR
- ☐ 1-year $ 8
- ☐ 3-year $21

INDIVIDUAL STUDENT
- ☐ 1-year $ 6
- ☐ 3-year $16

FOREIGN POSTAGE
- ☐ $1.50 per year

BILLING INSTRUCTIONS
- ☐ Payment enclosed
- ☐ Please bill me (Receive 1 issue less)

Name ______________________

Title or Dept. ______________________

School or Company ______________________

Street Address ______________________

City __________ State __________ Zip __________

Return form to: CREATIVE COMPUTING, P.O. Box 789-M, Morristown, New Jersey 07960

Neither atavism nor prettification will cope with the ecologic crises of our time
— Lyn White

It can be argued that development of any kind is essentially a learning process and that it is primarily dependent on a network of information flows
— Kenneth Boulding

Nations are such an artificial construct from an ecological point of view that any further energies poured into them are almost certain to do more harm than good.
— Keith Lampe

CREATIVE COMPUTING
P. O. Box 789-M
Morristown, New Jersey 07960

BULK RATE
U. S. POSTAGE
PAID
PERMIT No. 153
Morristown, NJ