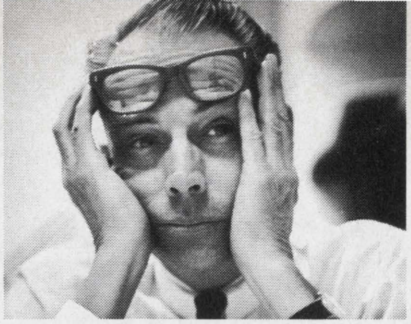
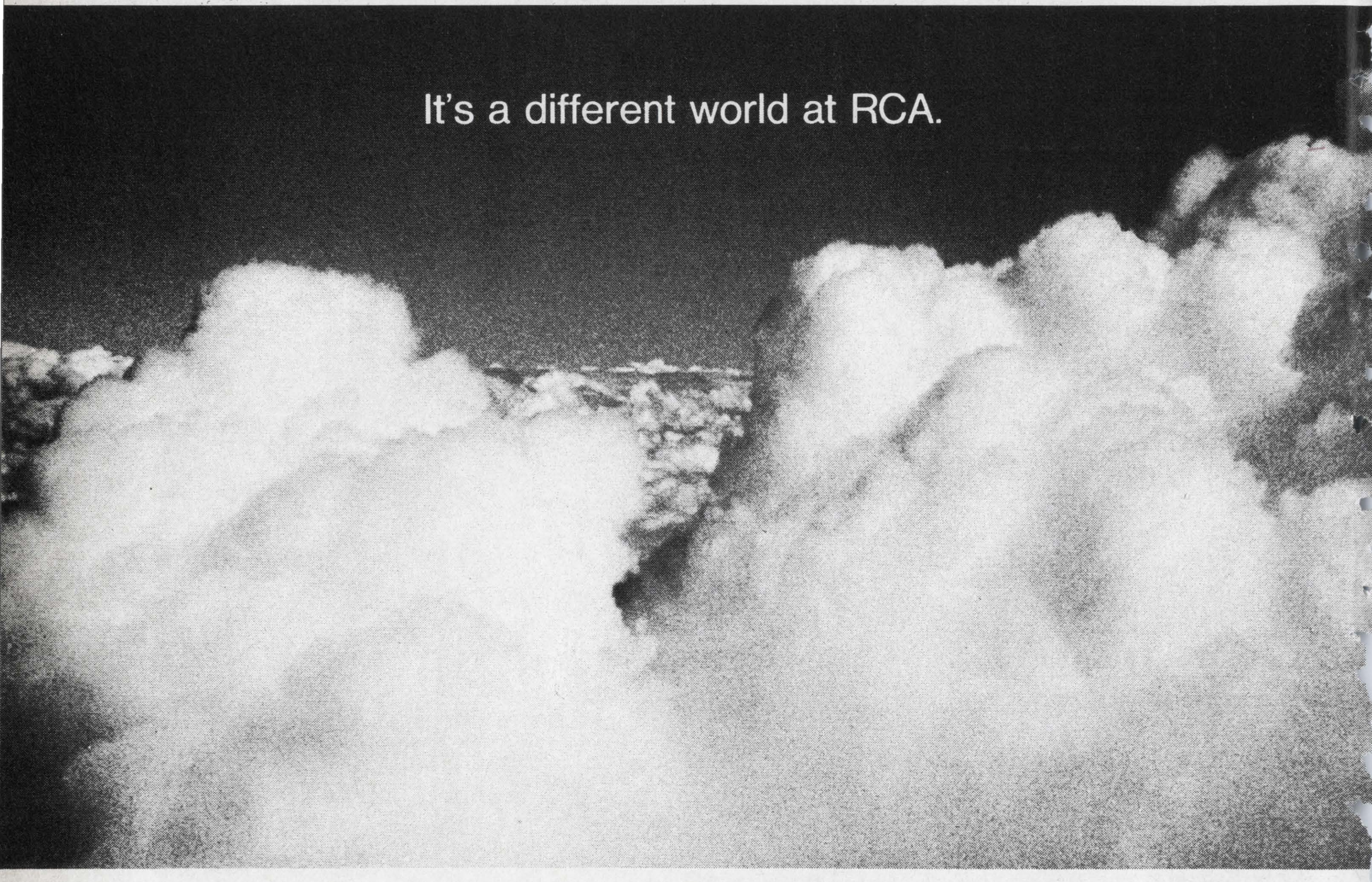
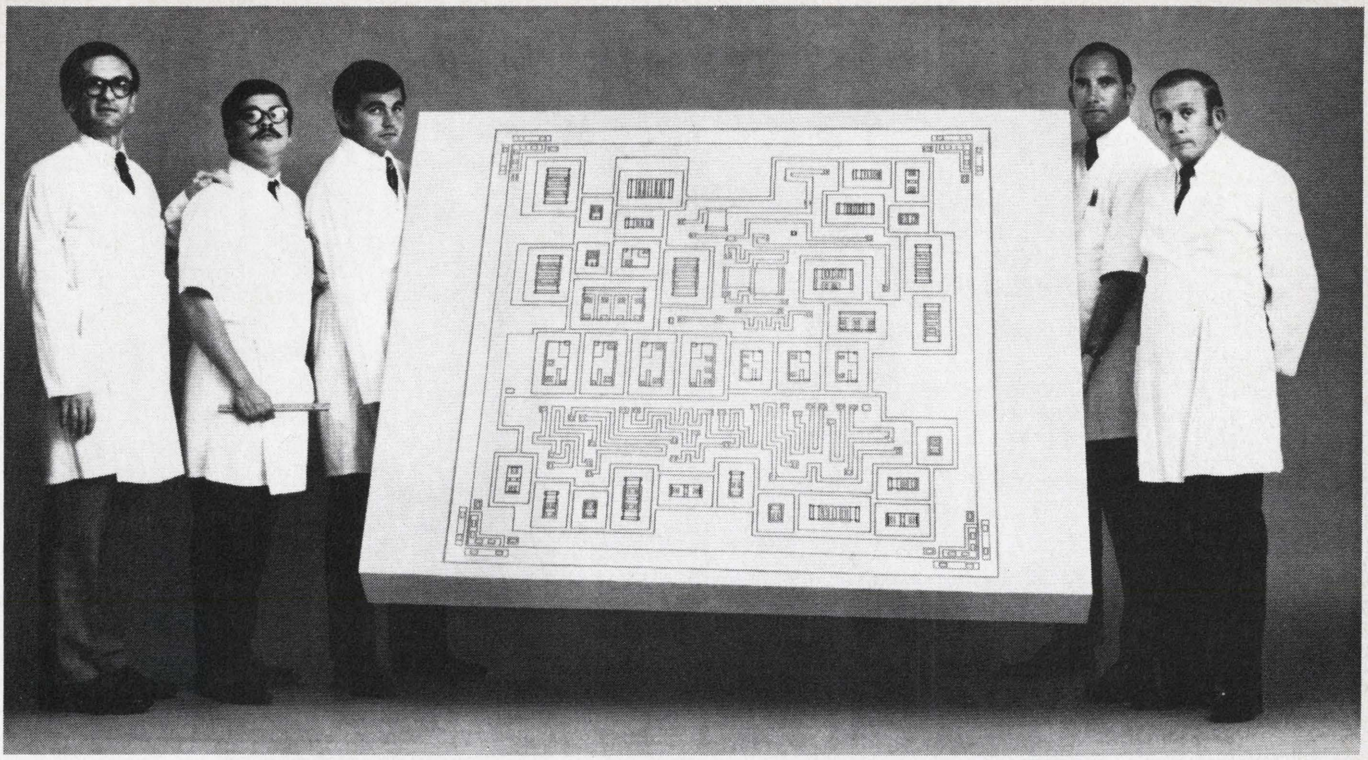
software age

JULY 1969




It's a different world at RCA.



software age

JULY, 1969 Vol. 3–No. 7

Publisher .... David W. French
Editor .... John F. Sikora
Managing Editor ...... Jean S. Logan
Editorial Assistant ...... Elspeth Daly
Advertising Manager .. John F. Neiman
Traffic Manager ..... Betsy Pavkovich
Circulation .... Ann Vickers
Business .... Stephenie Jensen
Resume Control .... Anne Wildeman

CIRCULATION OF THIS ISSUE
MORE THAN 130,000

SOFTWARE AGE
2211 FORDEM AVENUE
MADISON, WISCONSIN 53701
PHONE: (608) 249–0128
OR (800) 356–8141

John F. Neiman
New York, N. Y.: (212) 697–5356
Los Angeles, Calif.: Richard Faust,
9800 S. Sepulveda Blvd. (213) 776–0100
San Francisco, Calif.: Ernesto Montano,
595 Buckingham Way, Suite 226 (415) 731–8240
Philadelphia, Pa.: Richard D. Clemmer,
27 Acoma Lane, Collegeville, Pa. (215) 489–9141
Boston, Mass.: (617) 542–1466

SOFTWARE AGE
is published monthly by
press tech inc.
2211 Fordem Avenue,
Madison, Wisconsin 53701

Subscription free to qualified readers. Others, $10/yr. Individual copies, $1. Foreign subscriptions, $15/yr.

For change of address please return your address label from the magazine.

Controlled circulation postage paid at Madison, Wisconsin.

financial currents

A noteworthy projection emanating from the Diebold Research Program study indicates that computer terminal costs are expected to fall off some 40% in the next four years.

Reason given is threefold, says DRP.

1. Costs for integrated circuitry and core and magnetic film memory will decrease as much as 99% and 80%, respectively, or from one dollar to one cent per bit of information, and from five cents to one cent, again respectively. DRP sees lower costs here as a spur to more and better services for remote terminal users from a centrally located computer facility.

2. Growing competition among new terminal makers will mean cost benefits, bringing terminal prices down by 15% in the next few years through low overhead and marketing costs.

3. New display technology is expected to reduce cost of display devices by one-half in the coming three to five years. Perhaps—says DRP Associate Director Ernest von Simson—by mid-70's one of the new technologies now in the laboratory stage will be commercially available and will replace the cathode ray tube display as the most popular user output device.

All in all, communication costs will gradually decrease over the next decade, notes DRP, with the introduction of more sophisticated, reliable and cheaper equipment and with increased speeds of telephone-line data transmission. Over the short haul, buffer and message concentrators will probably have a more significant effect.

* * *

More than 40 basic units and models are included in a new line of miniature electronic instrumentation—dubbed VutroniK—from Honeywell's Industrial Division.

Made in U.S. and Glasgow, Scotland, the instrumentation is intended to give Honeywell a bigger slice of the $750-million a year American process controls market—a market which is expected to double and then some by 1980.

* * *

New companies, mergers, acquisitions:

Newly organized Iomec, Inc., Santa Clara, California, will specialize in total systems approaches to the problems of data storage, introducing as its first product Series 1000 Storage Subsystem—an online mass memory system designed for small to medium size computers. Four of five founders of Iomec are former IBM executives . . . The Interpublic Group of Companies, Inc., New York, and Informatics, Inc., Sherman Oaks, California, have completed negotiations for joint operation of Dataplan Inc., New York, with ownership at 70–30%, respectively. Annual revenues of Dataplan approach $2 million . . . Stock transfer will result in Computer Time–Sharing Corporation (CTC) acquiring Stockton Data Processing, a center providing commercial data processing services to customers in the San Joaquin Valley area. Recent acquisitions or agreements in principle by CTC represent 1968 earnings of about $300,000, on sales of approximately $4,500,000 . . . Computer Applications Incorporated, which operates seven data processing service centers throughout the country, will form a new corporation to carry out its dp activities. About 20% of its stock will be offered to the public . . . A new subsidiary corporation to provide time-sharing services has been formed by Pryor Computer Industries of Chicago. Headquarters will be at 400 North Michigan . . . UCC Financial Corporation is the new name of Gulf Group, Inc., wholly-owned financial subsidiary originally formed by University Computing Company in December. ■

TRIC Handles Corporate

Peter H. Vaughn, The Travelers Insurance Company

Since a great part of the activity in an insurance company is devoted to the acquisition, storage, and retrieval of information, developments in the field of Information Retrieval are clearly of significant interest to the insurance industry. While work in the field is being done by universities, software houses, and manufacturers, the magnitude of our information problem is such that we cannot afford to wait for a solution to materialize at some uncertain future date. Hence, resources are now being committed by The Travelers toward a solution, though it is clear that the current state of the art, with respect to hardware, precludes a total solution. Nevertheless, a dominant design criterion is the desire that our interim solution be a miniature of the long range solution, which will fill our total need when it can be placed in a hardware configuration adequate to the task. As if that were not enough, we have at least one more big problem: Corporate requirements for a total information system are a good deal less than precise. Keeping these and other constraints in mind, a prototype information retrieval system has been designed and is currently being implemented at The Travelers for the IBM 360/65. It will be observed that a prominent feature of the design is the facility to change our minds with relative ease, which is crucial so long as we are unable to specify exact requirements.

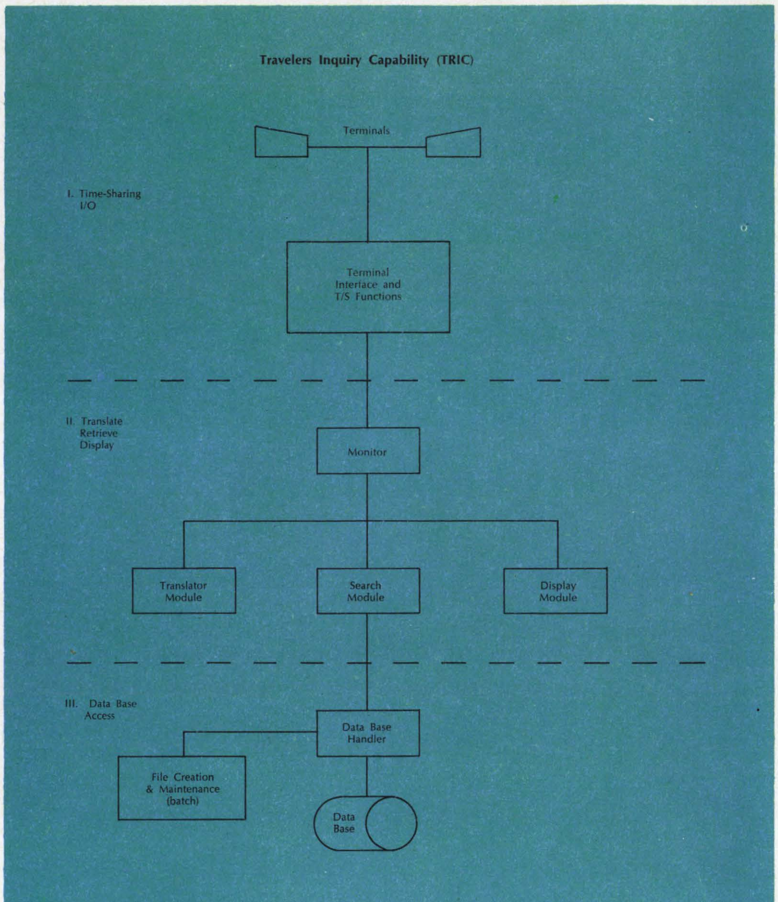


Figure 1.

In Figure 1, note that we have broken the system into three major subdivisions: (1) I/O and Time-sharing; (2) Translation, Retrieval, Display; and (3) Data Base Access. For most third generation main frames, time-sharing I/O and random access file handlers are available from the manufacturer. While these packages may be less than ideally suited to our needs, the modular design of the system permits us to use them in the early stages of development and replace them at a later time with "better" modules. Functional independence of modules is extremely important; procedural dependence of one module on another will be avoided, wherever possible (a change in the file handler will surely have an effect on the search module). Since we intend to use manufacturers' packages for I/O and file handling, our attention will, then, be focused on the middle area of the design: Translation, Retrieval, Display. It should be noted here that file creation and maintenance will be handled in the batch mode for the present, though it will be seen that expansion of the language to include update functions in entirely possible in the future. The balance of this article will be devoted to the various modules of the middle area and a brief discussion of the features of the initial version of the retrieval language.

Monitor

One of the most significant features of the gross design is the use of a monitor to control the flow of information through the system. When the monitor receives an inquiry across the time-sharing interface, it controls the sequence of calls to the translator, search, and display modules. When it becomes necessary to alter the flow or to add new

Information Retrieval

modules, only the monitor needs to be adjusted. All other modules are called, execute their specific functions, and return control to the monitor. It is this device which permits such a high degree of functional independence between modules. It is the monitor which is responsible for execution of administrative commands, such as attaching files for inquiry, and execution of user security validation prior to access to a given file. While it is possible, in an environment of well-defined requirements, to implement the control function of the monitor as a transition matrix, it was deemed inadvisable to do so in a situation where the probability of change approaches unity. Hence, the monitor is procedural code, and is, in fact, the entity with which the user is effectively in contact.

Retrieval Language

The rationale of the language is dialogue with a computer system; which is to say thus, we are essentially dealing with a question/answer environment. In general, the user will pose a question or input a command to the system via a terminal and receive an answer or an acknowledgement that his command has been executed. The converse of this, in which the system poses the question and the user answers it, permits us to control access to the system or to selected parts of the data base through identification and security procedures. This latter aspect of the interface is much less complex than the former and will not be treated here, since the problems involved are largely procedural in nature. Note that in either case, action by one side of the system must be met by action on the other before the dialogue can continue. That is, once the user has sent a query to the system, he is inhibited from further action until the system responds; likewise, once the system has displayed a response, nothing further happens until the user initiates another cycle or disconnects. It is this definition of the term "dialogue" which is intended, and not some artifice whereby the computer formats the reply into English language sentences.

Basic Format

The basic format of a user inquiry is:

DISPLAY : CONSTRAINT

In general, DISPLAY is what the user wants to see; CONSTRAINT is the conditions he wants imposed on records prior to the display.

Display

A display string may consist of one or more of the following, separated by a comma.

1) Name of one or more items, such as:
   NAME, ADDRESS, AGE
2) Name of a specific report, such as:
   REPT (SALARY WITHHOLDING),
   REPT (WRITTEN PREMIUMS)
3) Algebraic operation on items before display, such as:
   (GROSS − (FICA + BONDS + INC + PRKNG))
4) Certain functional commands, such as:
   MAIL (P. SMITH, CORP. SYS., NDC)—this would cause the report to be printed and mailed to the address in parentheses instead of being displayed to the terminal.

Constraint

A constraint string is a logical statement of the conditions which a record must meet if it is to be processed for display; it follows closely the mathematical notations encountered in algebra. The following list of operators is fairly complete, but it will be seen that expansion of the list would be quite simple.

1) relational operators $< \leq = \neq > \geq$
2) arithmetic operators $+ - \times \div$
3) boolean operators $\wedge$ *(intersection)* $\vee$ *(union)*
   (the negation operator, $\neg$, is not required since all cases are covered by the relational operator set)
4) alpha literal quote marks (")
5) parenthetical notation
6) functional notation *(use brackets for argument)* such as SUM, COUNT, MIN, MAX

Whenever a query is submitted which has no constraint portion, all records containing DISPLAY items are accepted for display. The following examples may help to clarify the use of the language. Note that all the item names that are used would be defined for the file being interrogated, and are distinguished from alpha literals by the absence of quotation marks.

1) EMP NO, NAME: WKLY SAL > 200
   A list of employee numbers and names for persons whose weekly salary is greater than $200.
2) POLNO, NAME, COVERAGE: ANN PREM > 10000
   A display of the policy number, policy holder name and coverage code for any policy on which the annual premium is greater than $10,000.
3) TITLE, AUTHOR: DESCRIP = "ATOMIC" $\vee$ DESCRIP = "ENERGY"
   A display of the title and author of any document for which one of the descriptors "atomic" or "energy" is listed.
4) REPT (OPERATIONS): DATE = 671231
   A display of a report named OPERATIONS, accepting data from records bearing the date Dec. 31, 1967.

Programmers and systems analysts...

Univac: Where you can help make the world a little bit better

An American astronaut hasn't walked on the surface of the moon . . . yet.

But now America's manned Apollo 8 space shot, with the help of UNIVAC® computers, has successfully orbited the moon. Which brings an American lunar landing that much closer to a reality.

The people who are proudest of this Apollo flight aren't just NASA people. They are also the Univac programmers and systems analysts who spent months writing the programs that made the use of telemetering of this flight possible.

You see, we understand that nothing bothers a programmer or systems analyst more than doing busy work. So we keep our people busy with meaningful work like the Apollo program.

What's more, we feel that the important problem is a problem that still hasn't been solved: the problem of keeping America strong. Because when America is strong she can better fulfill her obligations to underprivileged countries looking to her for help and direction. This is where you come in.

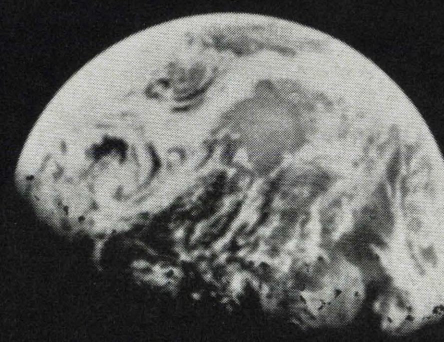
Univac needs people who are old-fashioned enough to care about making the world a little better. Who are interested in making a real contribution, not just marking time. And who want to be rewarded on the basis of the excellence of their service, instead of just on the length of their service.

If you're this kind of person, we here at Univac can't offer you a job. But we can offer you a fine career. Think it over and then check the facing page for a complete list of current Univac job opportunities.

SPERRY RAND

UNIVAC

An Equal Opportunity Employer M/F



5) MAIL (P. SMITH, CORP. SYS, NDC), REPT (OPERATIONS) : DATE = 671231
The report indicated in 4 above would be printed and mailed to the named individual instead of being displayed on the terminal.

6) REPT (CUR LIFE IN FORCE)
A report of the indicated name, accepting required data from any records, since no constraint is indicated.

7) SUM (WKLY SAL) : DEPT = "DATA PROC"
A summary of the weekly salaries in the Data Processing Department.

8) PREM — (LOSS + DIV + UPR + CONS + INS + TAX + EXP) : LINE = "CP" ∧ DATE = 671231 ∧ TITLE = "OPNS"
The indicated item values will be extracted, the indicated operations performed, and a single value will be displayed for any record in which the given constraints are met.

9) EMP NO, NAME: (WKLY SAL — FICA — TAX > 200
A list of the employee number and name for persons whose take-home weekly pay is greater than $200.

10) NAME: ((WKLY SAL — FICA — TAX) > 200) ∨ ANN SAL > 15000
A list of the names of weekly salaried people who take home more than $200 or annual salaried people who earn over $15,000.

11) POLNO: COUNT (CLAIM) > 3
A list of the policy numbers for which more than 3 claims have been recorded.

Key Words & Synonyms

The basic form of a query string is minimal. In order to make the language easier for non-technical personnel, the use of certain key words and synonyms for relational operators has been adopted. Use of the key words effectively labels the two parts of an inquiry; the current list includes the following:

Display	*Constraint*
DISPLAY	WHERE
SHOW	FOR
	WHEN
	IF

Synonyms for relational and boolean operators follows:

Symbol	*Synonym (bracketed by blanks)*
<	LT
≦	LE
=	EQ
≠	NE
>	GT
≧	GE
∧	AND
∨	OR

This second table turns out to be more a necessity than a convenience, since many keyboards do not have the mathematical symbols. Using the above-listed key words and synonyms, query # 1 may be re-stated:

IF WKLY SAL GT 200 DISPLAY EMP NO, NAME.

Translator

Since the execution of the procedures described herein is contingent upon the use of the information "bead", some preliminary description of beads is necessary. A bead is a contiguous collection of computer words or characters, containing pointers to other beads, which describes some piece of information. We note two essential characteristics of a bead; first, it is contiguous and may be treated as an entity; second, it contains pointers to other beads. It is this second characteristic from which the bead derives its name; the assignment of a memory address to a pointer field "strings" the bead on a particular chain, changing that value moves the bead to a new position or a new chain. Other information in the bead will be values which are manipulated by the programs using the beads. The contents of the bead which will be used in the translation procedure are as follows:

1) Bead type code
   a) operator
   b) item name
   c) function
2) Bead value
   a) direct storage for operators
   b) pointer to actual value for others
3) First pass pointers (forward and backward)
4) Upward pointer
5) Left operand pointer (down)
6) Right operand pointer (down)
7) Truth flags

By storing pointers for bead values, all beads can be made the same size without restricting the size of item or function names.

First Pass

The first pass through the constraint is merely a left-to-right scan, during which all of the individual elements of the character string are isolated and set into beads. These beads are strung together serially, using the First Pass Pointers. Thus, the expression "A > (B + 2)" is converted to a chain of beads which may be diagrammed as follows:

A—>—(—B—+—2—)

It is within this pass that certain notational conventions may be accommodated. For instance, unary operators are often a desirable feature. The value —7 is actually an algebraic shorthand notation for the expression O — 7. In order that the (+) and (—) operators may be processed consistently every time they are encountered, the first pass may replace the empty left operand of a unary operator with a constant zero. Thus, the expression "A > — 3" will be converted to:

A—>—O—−—3

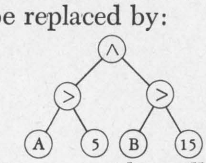
Another desirable convention is the grouping of operands under a common relational operator through the use of parentheses. In this case, the expression "A > (B ∨ C ∨ D)" will be replaced with:

A—>—B—∨—A—>—C—∨—A—>—D

By thus replacing shorthand expressions with the actual implied expressions, the processing of any operator becomes invariant, a fact which becomes important in constraint satisfaction.

Second Pass

The second pass may actually consist of several sub-passes, depending on the contents of the character string. It is the function of this pass to convert the single level chain of pass 1 into a structural diagram of the constraint, using the Upward, Left, and Right pointers. Thus the string A—>—5—∧—B—>—15 should be replaced by:



This is commonly called a tree structure; the reason for converting the chain to such a form will, again, become evident when constraint satisfaction is discussed. As the structural pointers get set, the length of the first pass connector chain gets smaller and smaller, until only one bead remains on it. At this point, it is no longer a chain and translation is complete, (see constraint examples 1 and 2).

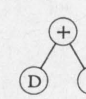
The scan proceeds forward on the first pass chain, counting open parenthesis, (, and open brackets, [. Whenever a close parenthesis,), or close bracket,], is encountered, the scan is interrupted and several evaluation passes take place. The evaluation passes take place between the last (or [and the one just encountered, (the symbols have to match, that is, ( ) or [ ]). The string between the parentheses is an expression, which is structurally evaluated and replaced in the first pass chain with a new form (see examples).

Evaluation

The rules for evaluation within ( ) and [ ] are identical. Since certain operators "outrank" others, several evaluation passes have to be made in order to be sure the operators are evaluated in the proper order. Evaluation of an operator consists of assigning its left operand (LO) and right operand (RO) pointers. The order in which operators are evaluated is as follows:

1) (exponentiation)
2) × ÷ /
3) + —
4) relational operators
5) boolean operators
6) , valid only within [ ] or ( )

Referring to constraint example 1, note the parenthetical expression in line 1. The only operator in the expression is (+); thus the evaluation consists of assigning the LO and RO for that operator, which produces:



The original 5 beads in the first pass chain are replaced with the single bead containing the (+) operator. This is done by merely adjusting the first pass pointer in the (=) operator to point directly to the (+) and the pointer in the (+) to point directly to the (/). The first pass pointers in the other beads, (, D , 3 ,), are not removed or changed, they merely become inoperative, since their structural pointers have now been set. The effective result is that the parentheses have been removed entirely and their syntactic contents have been represented structurally. After a given pair of parentheses has been evaluated, an immediate look to the left for a function name is executed. If the symbol immediately preceding the open parenthesis is a function name, the entire expression just evaluated is attached to its RO. After all parentheses have been removed one final evaluation pass takes place, which completes the translation. Proceeding in this manner has the effect of evaluating parentheses "from the inside out", which conforms to the way in which they are ordinarily used.

Search Module

The search module is concerned with two things: getting records from the file and testing them against constraints. Since discussion of file access necessarily includes discussion of a particular handler, it will not be useful to go into that aspect of the module now. However, constraint satisfaction is independent of the handler and merits some attention, particularly considering the final form of the translated string.

Using the final structural representation of Constraint Example 2 as a reference, we may begin by making some observations:

1) Item names and constant values always appear at the lowest level of any branch sub-structure, they are *never* nodes.
2) Nodes are either operators or function names.
3) All nodes may have some kind of value assigned to them, with the single exception of the comma, which is a "special case" pseudo operator.
   a) relational operators have 2

Figure 2. Constraint Example 1

$: A < B + 4 \wedge C = (D + 3)/5$

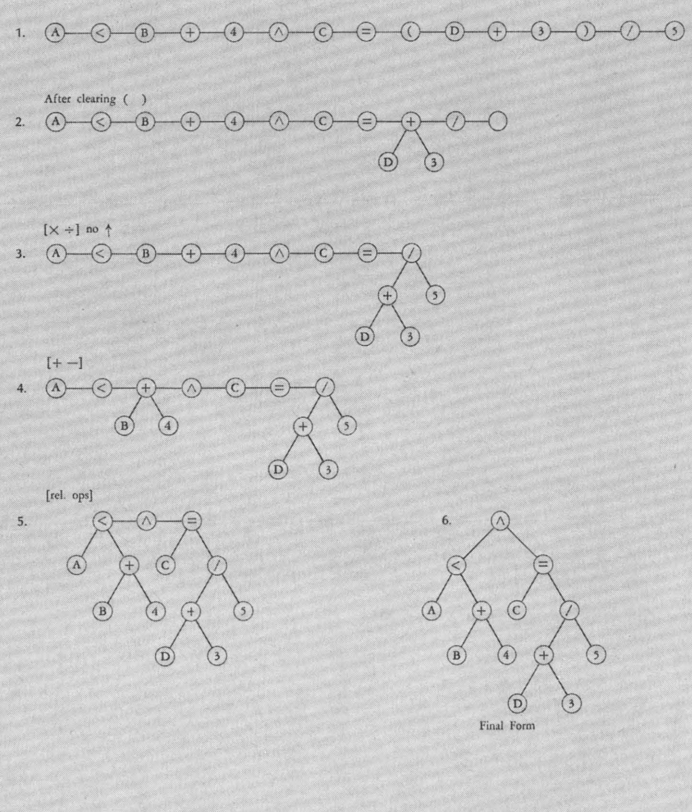
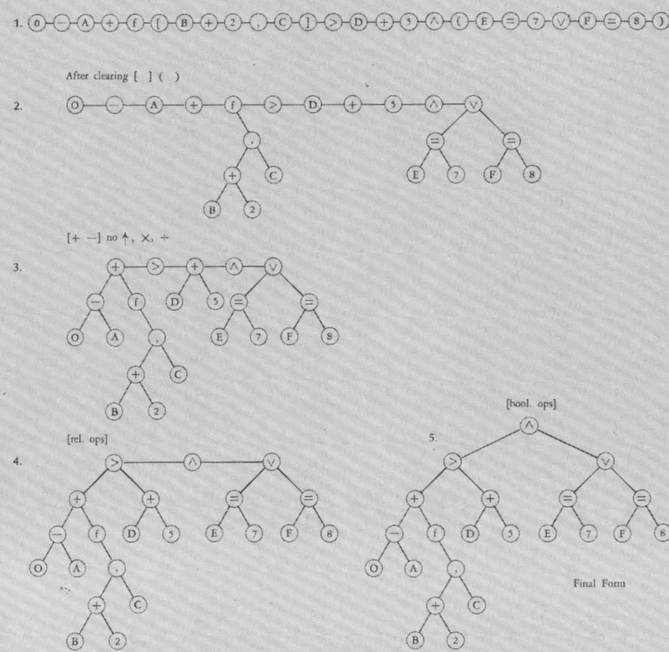


Figure 3. Constraint Example 2

$: -A + f[B + 2, C] > D + 5 \quad (E = 7 \quad F = 8)$



possible values: T (true) or F (false).

b) boolean operators can be T or F, but in the case of union (∨) each branch has to be considered separately.

Fundamentally, the answer to whether or not a record satisfies the constraint is found in the value of the bead which sits at the apex of the constraint. If the value = T, the constraint is satisfied. Since the assignment of a value to a node requires that the values of at least one, usually both, of its operands be known, the only place to begin this process is at the bottom of the tree, since that is the one place where we can assign values immediately.

Again, referring to the final diagram of Constraint Example 2, let us begin with the bottom of the leftmost branch. Moving up one level, we encounter the (−) operator, to which a value is assigned by subtracting the right operand (RO), in the case item A from the record, from the left operand (LO). We then move up one more level to the (+) operator; here we have a value for the LO, but not for the RO. In this instance, the RO is a function name, so after evaluating the expression (B + 2) in its argument string, we branch to the function f, which assigns a value to the node f. Now we can assign a value to the (+) operator and again move up. Here we have the relational operator (>), which requires us to compare the LO and RO, and, again, we have only the value of LO. Thus, we move to the bottom of the RO branch, in this case the node D, and begin the upward motion again, assigning values as we go. Once the value for the (+) has been assigned, we can evaluate the (>) and move upward. The apex operator is the symbol for intersection, which requires that both of its branches be true. Thus, if the value of the (>) is false, we may discontinue the test, since the constraint cannot be satisfied. If it is true its RO must be evaluated. Again, we move to the bottom and begin by evaluating the left most (=). Above that is the symbol for union ($\vee$), requiring only that one of its branches be true. Hence, if E = 7, ($\vee$) = T, and the constraint is satisfied; if not, we must test for F = 8. If both conditions fail, ($\vee$) = F and the constraint is not satisfied.

The foregoing narrative has presented all of the steps that would be executed, in their proper sequence. Observe, however, that the actual number of different steps is quite small:

1) Assign value to operator or function node.
2) Move up to next level.
3) If possible, repeat from step 1, otherwise, go down RO to bottom, then repeat from step 1.
4) If at apex, test for T or F.

Assignment of a value to an operator node consists of branching to a subroutine which executes only that operator; thus, addition of new operators is quite simple, in addition to having virtually no effect on the translator. The important thing to note here is that the constraint satisfaction routine is relatively small, it is simply iterated to its conclusion under the complete control of the tree structure.

Display Module

The display module, in its present form, is rather simple. There are two types of display: named reports, for which formats have been previously defined in a format library, and the default display. The default display is always used to display selected named items, and is merely a listing of attribute/value pairs, one per line, up to the screen limits. A query such as NAME, EMP NO : WKLY SAL > 200 might produce the following display:

```
NAME     ALLEN, JOHN P.
EMP NO   12377
*
NAME     JONES, WILLIAM J.
EMP NO   23443
*
NAME     WILLYS, JOANNE L.
EMP NO   44715
******
```

Note the single asterisk separating the information from different records and the final row of asterisks, indicating the end of the response. If that final row were absent, it would indicate that there was more information to display but no room on the screen. The user may summon the next block by signalling the computer from the terminal. Note that responsibility for recognition of the paging signal rests with the monitor, not the display module. The display module merely extracts display items from file records and puts them in a display buffer, which it then releases to the monitor.

Conclusions

Significant conclusions are premature at this point; our problem is large and multi-faceted. We have attacked an area that appears capable of solution, with the hope that our solution will not have to be discarded when we move into a new generation of machines, which we must surely do before we solve the whole problem. ■

Peter Vaughn *received a B.A. in Math (Summa Cum Laude) from the Evening Division of the University of Hartford in 1965. He has worked for Travelers Insurance since 1967 in man–machine interactive systems—specifically in computer aided design of integrated circuits and information retrieval.*

personal lines

Computer Communications, Inc.: **James D. Johnson,** Vice President, appointed to direct activity in corporate level staff functions which involves research, corporate development and technical evaluation of product and acquisition candidates. **Robert N. Windsor** appointed General Manager, CCI products, a new organization.

Computing and Software, Inc.: **Mits Tamura,** as Manager of On-Line Systems Development for the Western Computing Division, will be responsible for the development and coordination of comprehensive computerized information systems for industrial and business requirements and the internal needs of the corporation.

Cullinane Corporation: **Russell Varcoe** named Director of the Technical Services Division in New York and head of the firm's New York-based operations.

Data Processing Financial and General Corporation: **Dr. Joseph Robertson** appointed Vice President and Eastern Regional Manager of Information Systems Company, a division of Data Processing.

Dataram Corporation: **Joseph T. Brown,** named Manufacturing Manager, to be responsible for production, industrial engineering, purchasing, production testing and quality control for the Computer Products Division.

Informatics, Inc: **S. L. Benner** joins the Data Services Division as Manager of Insurance Applications, to direct sales and servicing of the Merritt System —an automated accounting system for insurance agents and brokers.

Information and Communication Applications, Inc.: **Paul M. Dangerfield** named Manager of Administration.

Information International: **James F. Gruder,** former senior project engineer, named Product Manager for graphic consoles, with responsibility for coordinating all marketing, research and engineering activities for the consoles.

Intranet Industries, Inc.: **Larry J. Patin,** former computer scientist, named Manager of the Diagnostic Programming Department of the Engineering Development Division. **Frank M. Stelle** advanced from systems programmer to Senior Systems Programmer in the Compiler Development Department of the Systems Development Division.

Leasco Systems and Research Corporation: **James H. Hough** joins the company as District Manager of the new Los Angeles office. **John J. Brooks** and **John D. Lipps** appointed branch managers of the Cleveland and Detroit offices, respectively.

Sperry Rand (Univac): **John P. Kiernan,** former Director, Research and Development, for Univac Federal Systems Division in Philadelphia, named Director, Planning and Operational Services, in the Data Processing Division. **Nate Pearlman,** former Manager, Special Products Marketing, appointed Manager of the newly formed Special Systems Group in Univac's Data Processing Division.

TROUBLE-TRAN PRESENTS
XTRAN'S ADVENTURES IN FORTRAN

By GEORGE N. VASSILAKIS

Send your ANSWER to the problems posed here in each issue to:

TROUBLE-TRAN EDITOR

software age

P. O. Box 2076

2211 Fordem Ave., Madison, Wis. 53701

You can also profit by submitting PROBLEMS for this feature. If your problem is FORTRAN programming is selected for use in this feature, you will receive **$25.00**

TROUBLE-TRAN's Objectives:

1. To have fun.
2. To promote USA Standard FORTRAN by pointing out differences and inconsistencies of existing FORTRAN Compilers.
3. To alert programmers to the physical limitations of hardware.

Contest Rules:

1. The best answer (best explanation) bearing the earliest postmark wins .. **$25.00**
2. The second best answer with the earliest postmark will net the reader submitting it **$15.00**

PROBLEM OF THE MONTH

Here is a different kind of a problem worth $50.00 to you, if your solution is postmarked prior to August 1, 1969, and is selected as the most efficient. Only one winner will be selected by a panel of three judges and their decision will be final.

Problem 15: Prime Numbers

Write a FORTRAN Program to compute all prime numbers from 1 to N using the following rules:

1. N is read from a data card and may have a maximum value of 10000.
2. Arithmetic operations are restricted to USA Standard FORTRAN and integer arithmetic.
3. Submit program listing and sample output with N = 500.
4. Count all arithmetic and logical operations and submit an estimate of executed operations for N = 10000. Count each addition = 1; subtraction = 1; compare = 1; multiplication = 2; division = 3; and ignore all other operations.
5. Total storage allocated to arrays should not exceed 1000 locations.

NOTE: Since I have to submit my material six weeks before publication time, I am forced to write the answer to last month's problem long before I see the problem or read any mail from TROUBLE-TRAN readers. So, if there were any typographical errors in last month's problem, please forgive me for not being able to apologize on time. And, when you see a little f in an IF statement, please don't write just to say "this program will not compile."

ANSWER TO LAST MONTH'S PROBLEM

Last Month's problem was stated as follows:

```
      DØ 10 I = 2,3
   10 IF (I.EQ.4) GØ TØ 20
   20 DØ 30 I = 1
   30 WRITE (6,90) I
   90 FØRMAT (4H1I = ,I4)
      STOP
      END
```

The statement in question was obviously statement 20. Is it a DØ statement or an arithmetic statement? The answer, of course, depends on the compiler. If we eliminate blanks and rewrite this statement as 20 DØ30I = 1, it is easy to see that this is an arithmetic statement and there is nothing wrong in using DØ30I as the name of a variable.

Compilers such as GE–635, IBM–7044 and many others will compile and execute this program treating statement 20 as a replacement statement.

The IBM–360 and other systems which do not allow blanks in the middle of a name will treat this as an illegal statement.

If the program compiles and executes, the value of I will be 3 if location I is updated at the beginning of the DØ loop. GE–635 updates in the beginning while the IBM–7044 updates at the end of the DØ loop.

XTRAN

SDS

WHERE THE ACTION REALLY IS

The Company An industry leader in the design and manufacture of computers and peripheral equipment for a broad range of applications – from real-time applications to medical research.

Our Growth From nine people in 1961 to over forty-seven hundred today.

If you are looking for a company where your career has some place to go ... SDS has the action ... and plenty of room for you to grow.

Application Software Product Managers Will be responsible for definition of software packages and project responsibility for their implementation including monitoring of external or internal development. Requires 6-10 years' programming experience in one of the following: simulation languages (GPSS), general business data processing, information retrieval, automatic numerical machine control and language processors.

Computer Applications Analysts Positions require customer liaison, supervision of software system installation, assisting customers in preparing special programs. Some travel required. Applicants must be familiar with all standard software packages. BS degree in math, engineering, or science preferred. Minimum of 2 years' experience required.

Language Processor Specialists Requires a minimum of 4 years' programming experience in one or more of the following: assembly languages, COBOL, FORTRAN, PL-1, SL-1, time-sharing, and/or real-time software systems.

Operating Systems Specialists Positions available in development or market support. Requires a minimum of 4 years' experience in medium- to large-scale monitor systems, including a knowledge of loaders, I/O queing, interrupt processing, and processor interfacing, and machine language programming in macro or meta assemblers. Experience in time-sharing also desirable.

Programming Instructors To teach state-of-the-art software. This is an opporttunity for the experienced systems programmer to broaden his knowledge and learn the latest in time-sharing, assemblers, compilers, and monitors for third generation systems. Students consist of company and customer programming personnel.

Engineering Programmers Will participate in the development of multilevel real-time maintenance programming systems, interfacing with both computer software and hardware development engineers. Successful applicants must have a strong background in software design and interest in hardware or a strong background in hardware design and interest in software. Technical degree required.

SDS
Scientific Data Systems
A XEROX COMPANY

Send resume to Mr. Cark Kundman

701 So. Aviation Blvd., Dept. SWA769/El Segundo, California 90245
An equal opportunity employer

conference countdown

AUGUST

4 IEEE Seminar on Cast Studies in System Control, Boulder, Colorado. Contact: Prof. J. B. Lewis, Electrical Engineering Dept., Penn. State University, University Park, Penn. 16802.

5– 7 1969 Joint Automatic Control Conference, Boulder, Colorado. Contact: W. E. Schiesser, Department of Chemical Engineering, Lehigh University, Bethlehem, Penn. 18015.

5– 8 The Genealogical Society's World Conference on Records, Salt Lake City, Utah. Contact: World Conference on Records, 79 S. State St., Salt Lake City, Utah 84111.

11–14 American Management Association's 5th Education and Training Conference, New York, N. Y., Contact: American Management Association, 135 W. 50th St., New York, N.Y. 10020.

19–21 Western Electronic Show and Convention (WESCON), San Francisco, Calif. Contact: WESCON, 3600 Wilshire Blvd., Los Angeles, Calif. 90005.

25–29 Datafair 69, Manchester, England. Contact: The British Computer Society, 21 Lamb's Conduit St., London W.C. 1, England.

26–28 ACM National Conference & Exposition, San Francisco, Calif., Contact: ACM, 1133 Avenue of the Americas, New York, N. Y. 10036.

SEPTEMBER

8– 9 Society for Management Information Systems First Annual Meeting, Minneapolis, Minn. Contact: G. W. Dickson, Management Information Research Center, School of Business Administration, University of Minnesota, Minneapolis, Minn. 55455.

8–10 American Institute of Aeronautics and Astronautics Computer Systems Committee Conference, Los Angeles, Calif. Contact: Dr. Eugene Levin, Aerospace Corp. P.O. Box 95085, Los Angeles, Calif. 90045.

8–12 International Symposium on Man–Machine Systems, Cambridge, England. Contact: Robert McLane, G-MMS Meetings Chairman, Honeywell Inc., 2345 Walnut St., St. Paul, Minn. 55113.

15–17 Joint Conference on Programming Languages for Numerically Controlled Machine Tools, Rome, Italy. Contact: E. L. Harder, R & D Center, Westinghouse Electric Corp., Beulah Rd., Pittsburg, Penn. 15235.

28– International Systems Meeting of the Association for Systems

OCTOBER

Oct. 1 Management, New York, N. Y. Contact: Richard Irwin, Association for Systems Management, 24587 Bagley Rd., Cleveland, Ohio 44138.

1– 5 American Society for Information Science Annual Meeting, San Francisco, Calif. Contact: ASIS, 2011 Eye St., N.W., Washington, D.C. 20006.

2– 3 Second Advanced EDP Audit and Control Conference, New York, N. Y. Contact: Harold Weiss, Automation Training Center, 1930 Isaac Newton Sq. E., Reston, Virginia 22010.

9–11 Developing Professional Manpower through Action, conference of DPMA Division 3, Little Rock, Arkansas. Contact: Robert Redus, 6901 Murray St., Little Rock, Arkansas.

19–22 1969 National Conference of the American Records Management Association, St. Louis, Mo. Contact: ARMA National Headquarters, 24 N. Wabash Ave., Chicago, Ill. 60602.

27–31 BEMA Annual Exposition & Conference, New York, N.Y. Contact: Paul Notari, BEMA, 235 E. 42nd St., New York, N.Y. 10017.

Reading ASCII Tape on a BCD Computer

by

Charles Erwin Cohn

Argonne National Laboratory, Argonne, Illinois

At the present time, the American Standard Code for Information Interchange (ASCII) is the standard for data recording on paper tape. However, there are still in existence many older computers which use 6-bit BCD paper-tape codes. The usefulness of such computers could be extended if they could be made to read ASCII tape. We have done that on a DDP-24 computer with a simple addition to the software and only a trivial modification to the hardware.

The DDP-24 paper-tape code uses channels 1–4, 6, and 7 for data bits, channel 1 being the low-order bit. Channel 5 carries the (odd) parity bit while channel 8 is used for a "stop code" or end-of-file mark. Since ASCII uses the DDP parity bit as an information bit, parity-error indications are used to assist ASCII translation and do not indicate actual errors. After an ASCII frame is read, the translation subroutine examines the status of the parity toggle to determine which of two correspondence tables is to be used to translate the character. (One of these is for even-parity characters, the other is for odd-parity characters.) The six data bits form a correspondence-table address, by which the equivalent BCD code is retrieved. A null character is returned for those ASCII characters not having a BCD counterpart.

In ASCII, channel 8 is sometimes used as a parity channel, but normally is always punched. However, the DDP-24 is wired to inhibit parity checking in any frame having channel 8 punched. (This was done so that a parity error will not be indicated on a frame containing only a stop code.) Therefore, for proper ASCII decoding on this machine, channel 8 must be disabled. That is done manually by a switch that was installed to ground the corresponding data line from the tape reader. This was the only hardware modification that had to be made.

The ASCII translation routine was coordinated with the normal FORTRAN–IV paper-tape input routine. To enter the ASCII mode, the statement CALL ASCII is executed. This causes a jump to the ASCII translation routine to be inserted in the paper-tape input routine, after which all input tapes are interpreted as ASCII tapes. To return to reading BCD tapes, the statement CALL DDPTPE is executed. This causes the jump to be replaced by a no-operation instruction. The translation routine, including the aforementioned entry points, is assembled separately from the paper tape input routine. Thus, it is loaded from the library tape only where needed, conserving memory space otherwise.

A number of modifications were made to the paper-tape input routine itself (in addition to a general tightening up). First, since the paper tape and typewriter input routines have most of their coding identical, they were consolidated to save memory space.

The original input routines were programmed for a fixed record length of 80 characters. If a carriage-return code appeared before 80 characters had been read, the record was terminated, with remaining characters set to blanks. If 80 characters were read without a carriage-return code appearing, the tape was run through until a carriage-return code was found. These characteristics made it impossible to read tapes produced by some of our laboratory equipment in which tape containing a great deal of data is produced with no carriage returns at all.

To eliminate these problems two modifications were made. First, the tape is stopped after a complete record had been read; it *is not* run through until a carriage-return code is found. (However, a carriage-return code encountered before a complete record still terminates the record.) Secondly, provisions are made for changing the record length. It is still initialized to 80 characters upon loading, but may be reduced by execution of the statement CALL SETINP(N) where N is the integer giving the new record length. (This new length then applies to typewriter as well as paper-tape input.)

When a stop code is encountered, it is treated like a carriage return. In addition, an end-of-file flag is set. The value of this flag may be retrieved by a FORTAN program through a suitable function reference. This provides the sort of input end-of-file test that is commonly available with the operating systems of large computers.

One of the tape formats that we have to read serves as a good example of the usage of these features. This BCD tape contains 1024 six-digit numbers, separated by blanks. The last number is followed by a stop code. The tape is read by the coding:

```
      DIMENSION ICHAN(1024)
      CALL SETINP(7)
      READ(2,1)ICHAN
    1 FORMAT(I6)
```

This coding reads a record of seven characters, consisting of one datum followed by a blank. The channel count is decoded by the I6 format and the blank is ignored. The process repeats until all the data have been read.

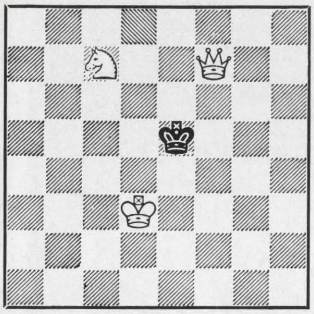
Our typewriter and tape-punch output routines also had most of their coding identical, so they too were consolidated to conserve memory space. ■

Charles Cohn *received a Ph.D. in Physics from the University of Chicago in 1957. Since then he has been at the Argonne National Laboratory as an associate physicist in the Reactor Physics Division. Cohn's major interest is the application of computer techniques to nuclear reactor physics experiments.*

CHECKMATE

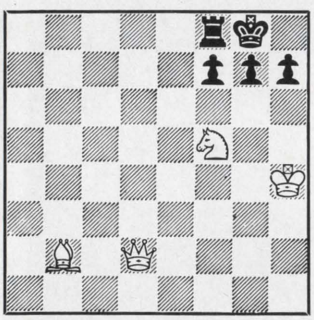
by
GEORGE N. VASSILAKIS
TRW SYSTEMS GROUP

Problem 15



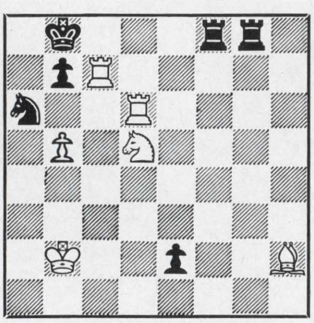
White Mates in Three

Problem 16



White Mates in Two

Problem 17



White Mates in Three

BERLINER's Chess-playing Computer

Hans J. Berliner, who holds the title of World's Correspondence Chess Champion, is now trying to teach a computer (IBM–360 Model 50) how to play chess. Hans works for IBM in Gaithersburg, Md., and spends his spare time working on a PL/I program which already plays a fair game of chess. Any computer which can last for 38 moves against two rated experts should be given at least a B rating.

Hans says the size of his program is 3000 PL/I statements, runs on the IBM 360/50 and may be accessed from the IBM 2250 Graphics Terminal, over remote teletype lines, or from the operator's console.

Here is the game the computer played against two rated experts, on April 21, 1969.

*Time (sec)		COMPUTER	EXPERTS	Time (sec)	*Time (sec)		COMPUTER	EXPERTS	Time (sec)
6	1.	P–Q4	P–Q4	10	69	20.	N–R4	Q–N4	19
9	2.	N–QB3	N–KB3	7	36	21.	N–N2	P–QR4	18
9	3.	N–B3	B–B4	7	17	22.	P–QR4	Q–R3	14
19	4.	P–K3	P–K3	6	17	23.	PxP	B–R6	5
13	5.	B–Q3	QN–Q2	7	26	24.	P–B5	BxN	40
5	6.	O–O	P–B3	6	16	25.	RxB	Q–QP	7
11	7.	R–N1	B–Q3	8	50	26.	RxP	R–K5	5
16	8.	B–Q2	Q–B2	11	47	27.	Q–R5	R–Q2	14
28	9.	P–KR3	O–O	27	38	28.	RxR	NxR	8
21	10.	P–KN4	BxB	17	43	29.	P–N6	N–B3	10
14	11.	PxB	P–K4	9	37	30.	PxBPch**	K–B1	18
33	12.	NxP	NxN	6	54	31.	Q–K2	QxQ	9
23	13.	PxN	BxP	2	26	32.	RxQ	RxRP	2
17	14.	P–B4	B–Q3	20	27	33.	R–N2	KxP	5
28	15.	P–N5	N–Q2	8	13	34.	K–R2	P–B4	9
28	16.	P–N4	KR–K1	24	26	35.	B–K1	R–R6	15
36	17.	Q–N4	N–B1	36	12	36.	B–Q2	N–K5	3
50	18.	P–QR3	QR–Q1	41	32	37.	R–K2 ?	R–R7	7
34	19.	R–B2	Q–N3	20	8	38.	R–N2	NxB	6

RESIGNS (time was up)

994 TOTAL 486

* Computer times are elapsed times in an MVT environment on the IBM 360 Model 50, not CPU times.

** On move 30 the computer could have obtained a clearly winning position by playing 30. PxRP ch, NxP, 31. R–N2, RxRP, 32. Q–R6, Q–N8ch, 33. K–R2, Q–N7, 34. QxBP.

Solution to Problem 12

Mate in ½ move, if you lift up white king and hold it in the air.

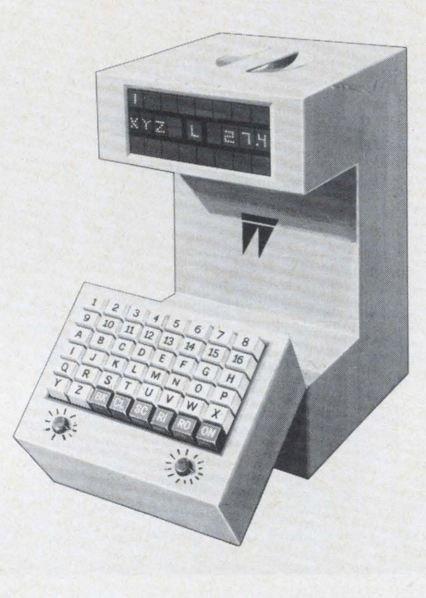
Solution to Problem 13

1 R–N7, P–Q4; 2 R–Q7, P–Q5; 3 R–Q6, P–Q6; 4 R–Q5, P–Q7; 5 R–Q3, P–Q8(Q); 6 R–R3ch, Q–R5; 7 R–R7ch, QxRch; 8 P–N7 mate.

Solution to Problem 14

Mate in 1: 1 P–K8(Q) mate
Mate in 2: 1 P–K8(R)ch, K–Q2; 2 R–K7 mate
Mate in 3: 1 P–K8(B), P–Q4; 2 K–B6, PxP; 3 B–Q7 mate
Mate in 4: 1 P–K8(N), K–Q2; 2 N–N7, P–Q4; 3 P–K5, P–Q5; 4 P–K6 mate

new products



The **TAPEWATCHER** is the first product of a new company, Tapewatcher Computer Corporation of New York. The "Tapewatcher", a small special purpose digital computer, is intended for use by brokers and investors who are active traders in the stock market.

The capabilities of the "Tapewatcher" include monitoring of all trades on the New York and American Stock Exchanges and notifying the user when variable and preset price or volume limits are exceeded by user selectable stocks.

The initial version of the "Tapewatcher" will watch up to sixteen stocks; will remember the last price, daily high, daily low and cumulative volume of all sixteen stocks; will alert user to "events" of interest and will display this information. In this phase of the system the "Tapewatcher" units will be connected to existing ticker networks such as those currently provided by the NYSE and AMEX.

For more information, circle No. 10 on the Reader Service Card.

The new Model 1051 **Data Terminal** from the Dura Division of Intercontinental Systems, Inc. is intended to bridge the gap between portable data terminals with dial-up capabilities and permanently located desk-structured, multi-unit terminals providing automatic data transmission.

The Model 1051, slightly larger than a standard electric typewriter, includes complete paper and edge card input/out capabilities, heavy-duty keyboard/printer, and a terminal control unit.

In addition to the convenience of portability, the Model 1051 enables the user to capture data on paper tape or edge card off-line, as desired, and then transmit the same data on-line, error free.

The 1051 Terminal is compatible with 360 systems and can replace 360 data terminals or operate in a mixed mode environment with them without any change in software.

The standard features include Non Print/Non Escape, which enable the operator or the computer to transmit or receive without printing for purposes of security. An automatic End-of-Line feature provides timing synchronization to prevent message loss during terminal-to-terminal transmissions. There are also several optional features available.

For more information, circle No. 11 on the Reader Service Card.

Synergistic Cybernetics, Inc. has developed a new proprietary software system, DEADLINE!, for **planning and scheduling computer center operations.**

As a planning tool the new system can be used to evaluate equipment and configuration alternatives and also to determine whether, under varying workloads, established deadlines can be met. Moreover, the system has the capability to evaluate staffing level, shift, and overtime alternatives, and to pinpoint imbalances and bottlenecks throughout the computer center.

The major features of the system are multi-processing scheduling capability, consideration of primary and alternate equipment availabilities, consideration of inter-job dependencies and inherent physical delays, leveling of machine and manpower requirements, and unique schemes for handling job priorities and undating schedules.

For more information, circle No. 12 on the Reader Service Card.

A new **high speed incremental plotter** has been introduced by the Graphic Systems Division of Computer Industries, Inc. It features adjustable plotting speeds of 2000, 1600, 1200 or 800 steps per second, program selectable step size s of 0.010, 0.005 or 0.0025 inch, and it is available with either 12 or 30 inch plot widths.

Designated Model 320 25, this new unit operates either on-line or off-line. Off-line operation is accomplished by either magnetic tape or punched card input. Each input option includes a Delta Control Unit utilizing input commands of up to 1024 steps per command and block transfer of the data for more economic plotting applications on-site or at remote terminals.

For more information, circle No. 13 on the Reader Service Card.

There is a new high performance **digital magnetic tape** unit on the market. Peripheral Equipment Corporation of Chatsworth, California produced the 6 x 40, and it has a speed of 37.5 ips and a read-after-write capability. It is designed for us with small and medium scale computer systems, in data terminal pooling and other mass storage applications, requiring high data transfer rates and immediate verification of recorded data.

The features of the 6 x 40 include a read-after-write dual stack head, operation speeds of 37.5 ips, 25 ips or 12.5 ips, a fast rewind speed of 150 ips, an adjustment-free tape guiding system, electronic deskswing and low power consumption. It is equipped with 10½ inch reels and can be ordered either in 9 track—800 bpi or 7 track-dual density configurations. The unit is compatible with IBM 729 and 2401 (Model 1) and IBM 2415 (Models 1–3 and 1–6). For applications not requiring the read-after write feature, a write/read single gap head series is available. Designated the 6 x 60, it provides the same high performance features as the 6 x 40 except for the ability to read-after-write. The 6 x 60 operates at 25 and 12.5 ips.

For more information, circle No. 14 on the Reader Service Card.

better books

by Dennie Van Tassel

COMPUTERS AND THE LAW. Edited by Robert P. Bigelow. Commerce Clearing House. 226 pages. 1969. $9.50.

This anthology is a thorough introduction to the state of the art in the use of computers by lawyers. Some twenty-seven computer-oriented experts have contributed their views about such areas as: Introduction to Machine Methods, The Computer in the Practice of Law, Government and the Computer, The Lawyer and his Client's Computer, and Jurimetrics (the science of predicting law results from past court action).

Martin Mayer comments in the foreword that this book consists of practical, essentially conservative papers. In law as in every field computers have been introducd initially to perform the most repetitive operations. In law, the main use seen for computing machinery, apart from the usual office work which law shares with everybody else, has been th retrieval of legal materials—statutes, administrative rulings and judicial opinions.

Law retrieval is one of the hardest jobs. It takes long hours of key punching to get the material into a computer since most law books are not printed so they can be scanned. Synonyms and synonymous phases must be added to make retrieval useful, but too many synonyms can cause such volumininous output that little work is saved.

Computerized law retrieval is just one of the areas which show great promise. Another more interesting and challenging area lies in the use of computer-based regressions, projections and simulations. Hypothetical legal arguments can be tested through use of simulation. Another similar use is in antitrust cases where both sides could use computer-based projections of the effects of a merger the U. S. government is trying to stop.

Soon new lawyers will have to be able to provide legal and technical arguments on the validity of computer simulation models in courts. Will these computer trained lawyers come from the computer or law professions? This book is so written that it is easily understood by non-lawyers.

DATA PROCESSING FOR DECISION-MAKING. By Richard W. Brightman. The Macmillan Company. 468 pages. 1968. $8.95.

The use of guess and intuition as decision-making techniques has seen its day. Modern information systems are able to provide decision makers with information, the quality of which was undreamed of a decade ago. But these information systems will be of use until management becomes familiar with their uses and potentialities.

This book is designed to help college students, managers, and executives understand the role that data processing takes within information systems and within the structure of the business organization.

Some of the topics covered are: Decision-Making Processes in Organizations; Scientific Decision-Making techniques; Information Systems; System Analysis and Design; Unit Record Information Systems; Programming; Operating Systems; Real-Time Systems and System 360 Assembly Language. Since only one fourth of the book covers information systems, the book seems most suitable first for the college student, and second, to the executive.

DEVELOPMENT OF INFORMATION SYSTEMS. By Donald F. Heany. The Ronald Press Company. 421 Pages. 1968. $9.00.

For information on how to correct some of the misconceptions that surround computers and banish the air of mystery which envelops computer based information systems, **Development of Information Systems** should be consulted. Computers have been the source of psychosomatic mi-

HELP WANTED

PROGRAMMER ANALYSTS, $9,000–$12,000. College degree plus minimum 2 yrs. programming exp., of which one year must be under 360 FULL OPERATING SYSTEM. Knowledge job control language and assembly language. Participate in development of a Management Information System, operating under a 360/40 FULL OPERATING SYSTEM. Will program utilizing all I/O devices incl: tape, disc, 2260's and 1050. **(201) 289–6000, Ext. 481.** Or send resume and salary history to: **Employment Manager, SEA-LAND SERVICE, P. O. Box 1050, Elizabeth, N. J. 07207.** An Equal Opportunity Employer.

EMPLOYMENT AND SEARCH COMPANIES

PROGRAMMERS 360 COBOL

Jrs. $8-10M, Srs. $16-22M, Intermediates $10-15M

We specialize in data processing placements. Resume not necessary. Company pays entire agency fee plus travel expenses.

ERIN EXECUTIVE AGENCY
132 W. 34 St., New York, N. Y. 10001
Phone: (212) 594–8296

DATA PROCESSING POSITIONS IN THE SOUTHWEST
Contact:
DATA PROCESSING CAREERS
Richard Kemmerly, Suite 1109, Stemmons Towers West, Dallas, Texas 75207. 214/637–6360.

E.D.P. SPECIALISTS — READY FOR A CHANGE? My client company is a major computer manufacturer who is looking for an unusual combination of ability: a young, aggressive, out-going, self-motivated man with one to three years Fortran experience who wants to reach the top quickly. You will represent this company in the L.A. area helping customers solve major problems. (If you want to be in top management in less than 5 years, this is for you.) Start at $10,000 to $14,000. With instant profit sharing, you can IN YOUR FIRST YEAR earn up to: $20,000. Contact E.D.P. Central, A Division of Employer's Central, Inc., Agency, 3600 Wilshire Blvd., Suite 1426 (213) 381–7662, Los Angeles, California 90005.

PENNSYLVANIA, NEW JERSEY, DELAWARE contain diversified opportunities for professional talent in all functions of E.D.P. If you desire to pinpoint your search in our area, contact us. With 9 highly effective years behind us we need not employ gimmicks nor shotgun duplicating techniques. We are large enough to have extensive knowledge of the market yet disciplined enough to remain personal and confidential. All fees paid. K. J. Gillespie, ANDERSON–TAYLOR Associates, 230 S. 15th St., Philadelphia, Pa. 19102. (215) KI 6–6333.

graines for businessmen for a long time.

Heany explains what is involved in designing, programming, and running an information system. He also stresses the important, indeed, indispensable contributions business executives can make to system development, particularly with respect to specifying what information is to be generated, how implementation is to be accomplished, and how systems work is to relate to operating work.

"In our June 1969 issue, we neglected to include the affiliation of Dr. Thomas C. Lowe, author of "Partially Predictable Results from Non-Deterministic Machines". Dr. Lowe is with Informatics Inc. where he serves as Manager/System Projects in the Washington, D. C. Division."

PROFESSIONAL SERVICES

ADD / SUBTRACT 6-DIGIT HEXADECIMALS in seconds with 100% accuracy. The pocket HEXADDER, $15. Free brochure from HEXCO, Dept. SA, P. O. Box 55588, Houston, Texas 77055.

CONSULTANTS: Commercial applications 1400, 360. Conversions, programming, systems design, testing, documenting. Job or contract. Computerized business mailing list, United States, $35 per 100 names. ALLIED COMPUTER CONSULTANTS, Box 133, Fairlawn, N. J. 07410.

PROGRAMMER TRAINEE: Mature, Business Administration AA graduate. Just completed six months total concept computer studies. Now available as entry level programmer. Previous potentially valuable experience—technical editing and writing. G. HIKADE, 4728 Lorelei, Long Beach, Calif. 90808.

WE WILL CONTRACT system planning, application, or system work to $10,000. We possess a complete background in statistics, linear programming, mathematics, civil engineering, processors, business data systems, data retrieval/processing; with a unique knowledge of OS/360 software and hardware. For more information in your field, write: SYSTEM PLANNING AND SOFTWARE MANAGEMENT, S/A Box 701, 2211 Fordem Avenue, Madison, Wisconsin 53701.

S/A market place

where you can reach 130,000 programmers, mathematicians, analysts and EDP managers

RATES

Classified—75¢ per word, minimum order $7.50. Blind ads are an additional $10.00. Payment must accompany copy.

Display Classified—One inch by one column, $75.00. Column width is 2¼". Help wanted and employment services ads will be accepted in the classified section. Ads of ⅛ page or more will be listed in the Advertisers' Index, and will qualify for the free resume service.

Closing Date—the 10th of the month preceding (i.e. May issue closes April 10th). Send order and remittance to:

software age
Classified Department
P. O. Box 2076, Madison, Wisconsin 53701

CLASSIFIED ADVERTISING ORDER FORM

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
26	27	28	29	30
31	32	33	34	35

________Words = $________

☐ Blind Ad $10.00 Insert________time(s) Total Enclosed $________

NAME________

ADDRESS________

CITY________ STATE________ ZIP________

Signature________

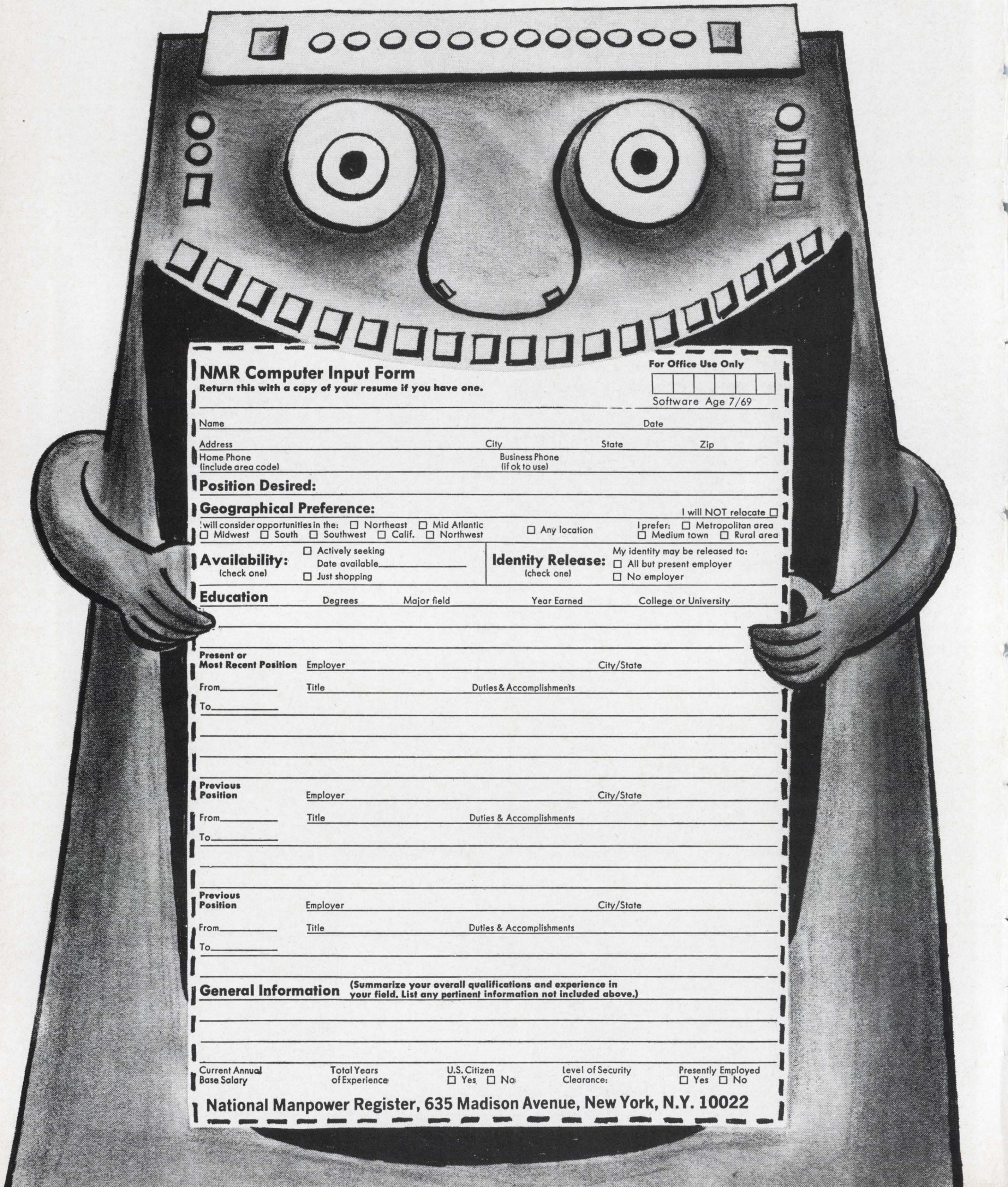
WORD COUNT: Includes name and address (unless blind ad). The name of the city (Des Moines), state (New York) and zip code are counted as a total of two (2) words. Each abbreviation, initial, single figure or group of figures or letters count as a word. Symbols such as 35mm, COD, AC, etc. count as a single word. Hyphenated words count as two (2) words.

We'd like to feed you to our computer.

Our computer has over 16,000 current, thoroughly screened jobs to fill with engineers, scientists, computer professionals and accountants. So with such a selection you just know one of the 56 nationwide placement specialists tied in to our system can locate a better job for you.

Naturally there's never any fee with NMR. And we only release your identity if we have your OK. So fill in this form and send it along to us with your latest resume.

Go ahead. Feed a hungry computer. NMR



For more information, circle No. 19 on the Reader Service Card

software age

CONFIDENTIAL INQUIRY

Your original copy of this form will be retained at the offices of SOFTWARE AGE and will be used for no other purpose than to notify the specific firms which you have checked (on the reverse side) of your interest.

THIS INQUIRY IS IN DIRECT RESPONSE TO YOUR ADVERTISEMENT IN SOFTWARE AGE MAGAZINE

TYPE OR PRINT CLEARLY FOR PHOTO REPRODUCTION

JOB DESIRED: ______________________________

List computer hardware knowledge (names of systems, tape, disk, terminals, etc.): ______________________________

Programming specialties and years of experience (commercial, scientific, theoretical, experimental, analog, etc.): ______________________________

Systems programming on which you have had development experience (compilers, assemblers, executives, monitors, O.S., etc. Indicate for what computer): ______________________________

Programming languages used and extent of experience (COBOL, FORTRAN, etc.): ______________________________

Applications programmed (aerospace, banking, insurance, math subroutines, compilers, etc.): ______________________________

Systems analysis experience (card design, flow charting, operation analysis, etc.): ______________________________

EDP management experience (include years and number of people reporting to you): ______________________________

SALARY: ______________ (current) ______________ (desired) **DATE OF AVAILABILITY:** ______________

EDUCATION: Indicate major as well as degree unless self-explanatory.

Degrees			
Years			
Schools			

EMPLOYMENT: Indicate present employment and previous jobs below.

Employer			
City			
Years	to	to	to
Title or Function			

Name ______________________________ Signature ______________________________

Home Address ______________________________ (city) Home Phone ______________

______________________________ (state) (ZIP code) Age ________ U. S. Citizen? ________

Security Clearance ______________________________ Location Preference ______________________________

Marital Status ______________________________

Military Status ______________________________

BE SURE YOU HAVE CHECKED ON REVERSE SIDE THE COMPANIES YOU WANT TO SEE THIS INQUIRY. PUT FORM IN STAMPED ENVELOPE AND MAIL TO:

software age
MAGAZINE
P. O. BOX 2076
2211 FORDEM AVE., MADISON, WIS. 53701

check your interests here

Fill in the confidential inquiry form on the other side of this sheet. This form provides all the information advertisers require to screen applicants. If further information is desired, you will hear from the advertiser direct. Then, check below the **boxes of those companies to which you want copies of your** form sent. Mail to SOFTWARE AGE, P.O. Box 2076, 2211 Fordem Avenue, Madison, Wisconsin 53701. (**Please do not send us your own resume. We will only process this form. A new form must be filled out for each issue in which you are answering ads.)**

EMPLOYMENT AND SEARCH AGENCIES

PRODUCTS AND SERVICES

(Use Reader Service Card)

☐ I do not now receive S/A. Please enter my FREE subscription.

....
Name Street Address

....
City State Zip Code

....................................../......................................
Prime Experience in What Industry My Specialty

☐ Technical Degree ☐ Non-Technical Degree ☐ No Degree

....
Year Born

☐ I Have Analog/Hybrid Experience

software age

P. O. Box 2076
2211 Fordem Avenue
Madison, Wisconsin 53701