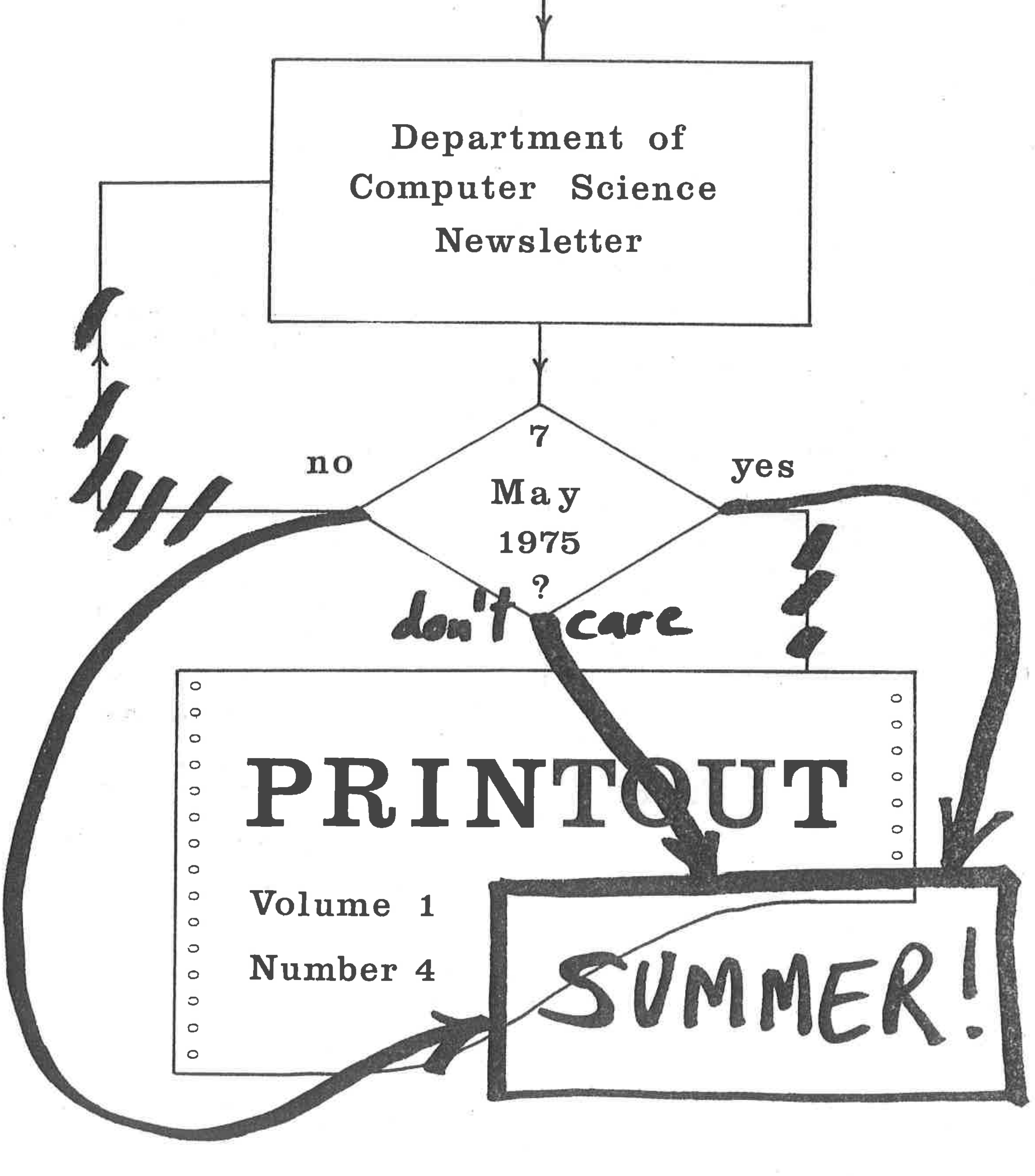
MATHEMATICAL and PHYSICAL SCIENCES and ENGINEERING
UNIVERSITY OF MARYLAND
1807-1856-1920
1973



Department of Computer Science Newsletter

7 May 1975 ?

PRINTOUT

Volume 1
Number 4

SUMMER!

PRINTOUT, the newsletter of the Department of Computer Science of the University of Maryland at College Park, is published sporadically and distributed to faculty, staff, and students in the Department. Opinions expressed in signed articles may be those of the author, but no opinions represent the policy of the Department, or of the College Park Campus, or of the University.

Contributions may be submitted to the editor, and unless they are obscene or seditious they will probably be used, but minor editing may be done. Complaints directed to the newsletter will be investigated and publicized when possible. It is well to keep in mind however that the Department is subordinate to higher levels of administration, not the other way around; and, the Department does not provide computing service to the campus. Complaints in these areas are best directed to other publications.

STAFF

EDITOR
DICK HAMLET

PRINTOUT

Volume 1
Number 4

CONTENTS

STUDENT ~~FACULTY~~ & COURSE EVALUATION

For several years instructor evaluation forms have been used by students in order to rate the various instructors in the department. Since several new faculty members will be joining the department next fall, it was felt that a similar form should be created in order to give these new faculty members advice on what students and courses to teach. The following is a first draft of this form which will shortly be submitted to faculty for approval.

- M. Zelkowitz, Chairperson
Ad Hoc Student Evaluation Committee

1. Course: CMSC ______

2. Instructor's rank:

(a) Instructor
(b) Assistant Professor
(c) Associate Professor
(d) Professor
(e) Research Professor

3. Why did you teach this course:

(a) Personal interest in subject matter
(b) Required by departmental chairman
(c) Easy course, leaves more time for research
(d) Class time fit my schedule well

4. Rate the text used in the course:

(a) Very good since I wrote it myself
(b) Poor, I should have written it myself
(c) No text available. I was too lazy to get my notes published

5. Do students take advantage of office hours?

(a) All term
(b) Before exams only
(c) Before finals only
(d) What's an office hour?

6. Were the assignments worth grading?

(a) Yes, got two published papers by posing research problems as trivial assignments
(b) Sometimes - used furthest-down-stairs-is- A approach
(c) No - live in rambler style house - no stairs

7. Did the course correlate well with catalog description of it?

(a) Course closely followed catalog description
(b) Catalog description out of date, water-powered Turing machines replaced by steam-powered.
(c) Don't know, never saw catalog description of course

8. Rate students handwriting:

(a) Quality of writing was inversely proportional to quality of answers
(b) Offered gold stars for neatness, but no takers
(c) Never read papers - let TA do grading

9. Class preparation time:

(a) Knew material from last term
(b) Managed to stay a lecture ahead of class
(c) Ran course as a seminar - let students do preparation

10. Computer project:

(a) Had class write an APL interpreter in PL/1 that was simulated by a SIMPL program for the PDP 11.
(b) All programs had to be HLL SPs. CISSes and DATs were expected to be used. MOLs were acceptable if V and V.
(c) Assigned programs for a hypothetical computer called the Univac 1108

11. How was class attendance?

    (a) An exam a day kept attendance up
    (b) The average fraction of students attending was approximately $1-X^2$ where X is the fraction of the course having elapsed
    (c) The professor who comes in 15 minutes late is rare; in fact he is in a class by himself

12. Was a teaching assistant helpful?

    (a) TA graded assignments
    (b) TA graded assignments and exams
    (c) TA graded assignments and exams and taught several lectures
    (d) TA graded assignments and exams and taught all lectures
    (e) What more can you ask for?

13. What sort of grades did you give?

    (a) All A's; after all the students have to get into graduate school
    (b) All A's; I wonder if giving exactly the same tests for the past seven years is educationally sound....
    (c) C was the highest grade; when I was a student, students really knew how to study

14. In summary, this course:

    (a) was a valuable review of material I should have know previously
    (b) is still totally incomprehensible
    (c) gave me deep insights into the true meanings of computer science and has awakened in me a desire to further investigate such aspects as has been made apparent by the knowledgeable studies of the other class participants.

IF $\frac{8}{0} = \infty$

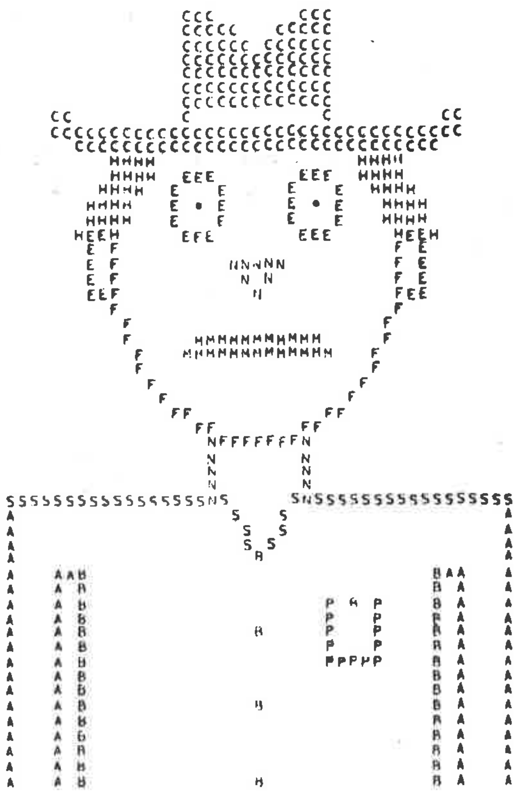
THEN $\frac{4}{0} =$

END

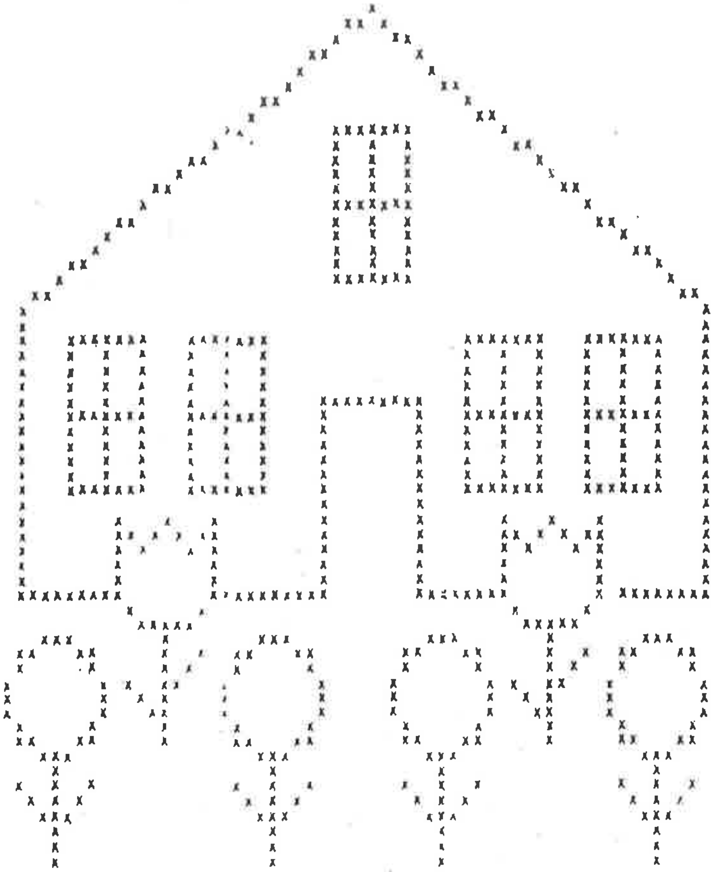
Non-numeric Programming

The line printer output reproduced below and on page 5 (all are about 2/3 actual size, Univac drum printer on 16 lb. standard stock, Anonymous, Not for Sale) arose in an assignment for CMSC 103 (Nagel).

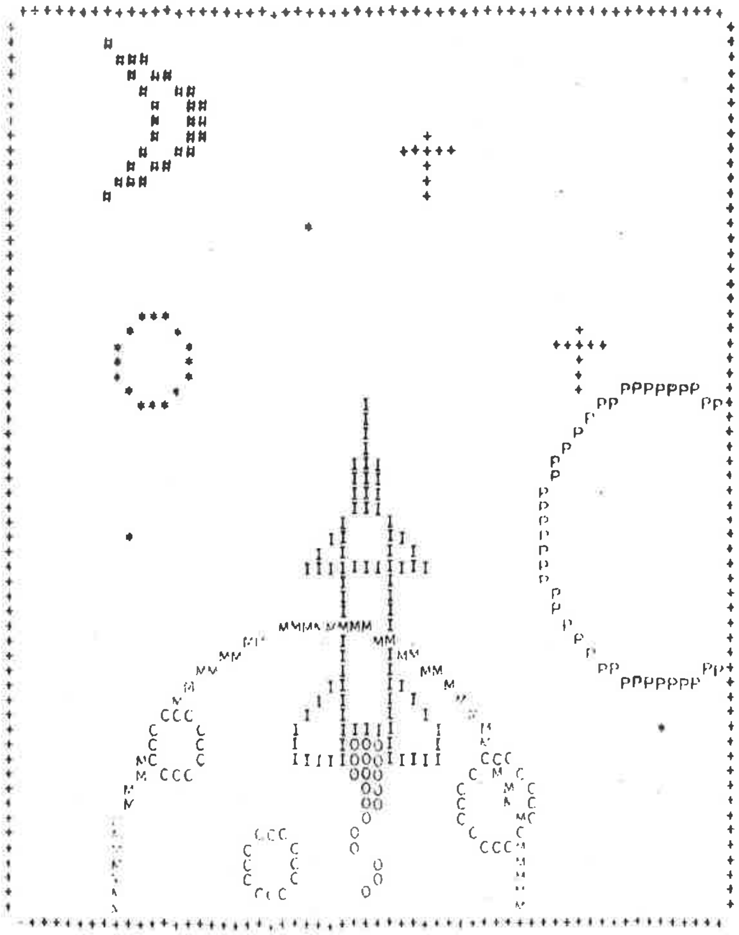
Students were given a fixed grid and routines for relocatable lines, circles, and points. They had to create a figure (below), and an original design (page 5).



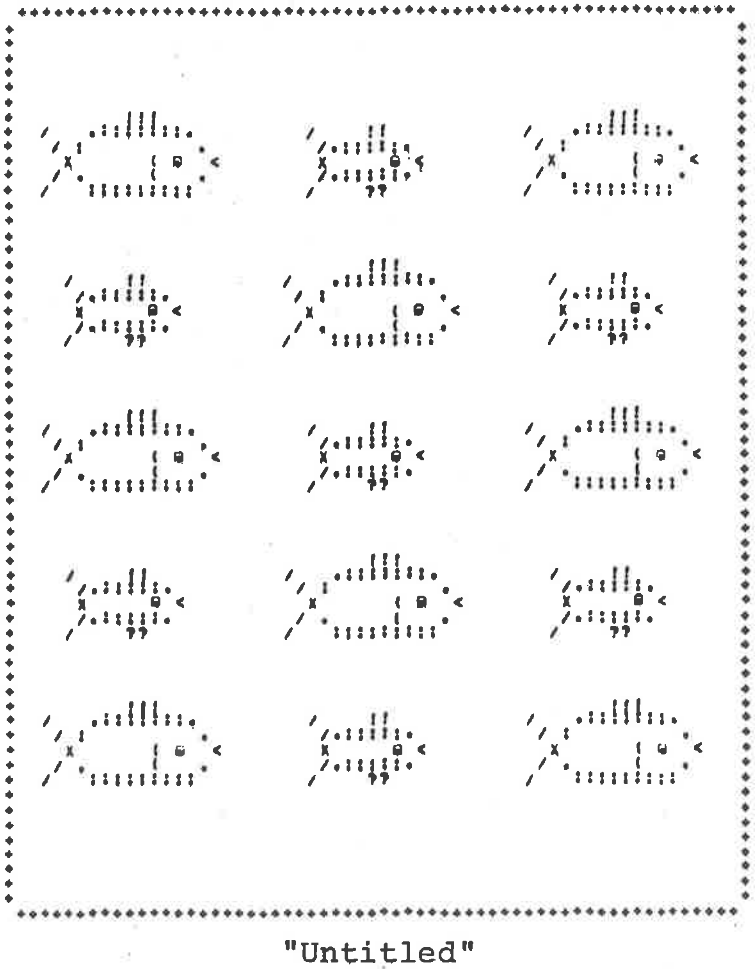
"Herman"
1975



"Untitled"
1975



"One Rocket Ship Crusing Past a Large Moon"
1975



"Untitled"
1975

A five-card deck whose cards are labeled "IDLE 1", "2", "3", "4", and "L/C" (on the back), and containing the following punches:

```
7
8 (overpunch)

$JOB            IDLE

$IBSYS

$ID   IDLE*000/65/000$

$STOP
```

was submitted to this Journal, rubberbanded to a note reading "a valuable piece of archival data." It is being held for the rightful owner.

FedUp

Dear Editor,

In PRINTOUT 1, #3 you noted that

```
REFETY ROWSETY ROWWSETY NONROW slice:
   weak REFETY ROWS ROWWSETY NONROW
   primary, sub symbol, ROWS leaving
   ROWSETY indexer, bus symbol.
   {R8.5.1a)}
```

This is perfectly true. What I take issue with is that printing this might turn some people off from Algol 68 and van Wijngaarden grammars. After all, let's not kid ourselves: what will the average reader think of A68 after reading the above production? He will obviously surmise: "Boo, another trivial language, another sleepy grammar." And yet this is not true! Those of us that know and cherish the Report know the joy of the sweeping crescendos, the crystal pianissimos, the stuning tutti...all I'm saying is why give such a simple example? Why not something grand like

```
LOSETY closed LMOODSETY LMOOD end BOX:
   LOSETY closed LMOODSETY LMOODSETY
   LMOOD end BOX; LOSETY open
   LMOODSETY LMOOD BOX. {R7.1.1.ee)}
```

or of course the simple but profound R8.2.Ø.1.b)

```
SORT coercend: SORT COERCEND.
```

Just remember the sleepless nights we all spent pondering the implications of this particular production on the future of mankind! This is the kind of inspiration our youth needs to embrace the teachings of the revered van Wijngaarden!

Peter Koves and Mark Stega,
on behalf of the Concerned Desciples
of Algol 68, UOM sect

You'll be pleased to learn that the Revised Report (Acta Informatica, to appear if A.vWg approves) has no sections 8.6.1a), 7.1.1ee), or 8.2.0.1b). Put that in your incense burner and ignite it. Ed.

Dear Sir:

It is incumbent upon the editor of a journal running a contest ["?s", PRINTOUT 1, #3] which offers a valuable price (such as the promised case of California wine in screw top bottles) to make the rules clear. In your journal you offer a problem, under the psuedonym of a M. Hecht, which needs a great deal of clarification.

The first problem is what meaning one can attach "in SIMPL-T" to the term temporary. The example has three equivalent variables A, B ,and T. A more significant problem is the mysterious function XOR which uses an unknown logical operator .XOR. (at least I can't find it in CN-14.1).

If the intent is to allow new primitives into the language which are known to be realized as computer operations on same machine, then one can just use the XCH operation which is modeled after the PDP-10's EXCH or the IBM-709's XCA.

If one is to infer the rules from the example given, then each contestant may give a different meaning to the problem. The only fair solution is to give each such contestant, or at least me, the prize.

R.G. Pythagoras

Computing has refused to send Donald Knuth their paper by airmail so that he could compete in the contest in which (corked) wine is awarded; they obviously feared the devastating affect he might have; as yours was the only entry we are pleased to announce the contest closed just prior to its receipt. Ed.

DEAR PRINTOUT,

(1) RECENTLY USERS SERVICES REQUESTED A NEW UNIVAC 1110, TO REPLACE THE PRESENT 1106. WHY WERE THEY REFUSED? WHAT IS THE PROCEDURE TO REQUEST A NEW COMPUTER? HOW CAN ANYONE FIND OUT WHAT'S GOING ON? MAYBE THE DIAMONDBACK WOULD LIKE TO INVESTIGATE AND COVER IT.

(2) DO MORE USERS OUTSIDE THE DEPARTMENT USE THE COMPUTER THAN THE USERS INSIDE?

(3) THE PROGRAM LIBRARY IS GETTING TOO BIG, ESPECIALLY SINCE ITS SIZE HAS NOT CHANGED SINCE THE UNDERGRADUATE PROGRAM WAS ADDED. MAYBE THE PROGRAM

LIBRARY COULD BE MOVED INTO THE SYSTEMS AREA SO THAT IT WOULD NOT BE AS BUSY, IF A SMALLER AREA COULD BE FOUND, AND SINCE THE SYSTEMS AREA IS NOT GENERALLY ACCESSABLE TO MOST USERS WHO NEED IT.

(4) MAYBE THE DIAMONDBACK WOULD COVER SOME ARTICLES AS TO WHAT IS WRONG, SCIENTIFICALLY, WITH HAVING THE UNIVAC HERE, AND WHAT WOULD BE BETTER WITH HAVING ANOTHER & DIFFERENT BRAND.

ANONYMOUS

(1) As noted on the inside cover, the Department is not responsible for the Univac systems, User Services, or the State of Maryland ADP Office. It is the latter that turned down the 1110 request, so Annapolis is the place to take complaints. For a state which spends so little on the education of its citizens (37th among 50 in per capita expenditure, for example), the 1106 is above par. (2) According to Computer Center Newsletter statistics, the 1106 is solidly used for classroom activity, of which the majority may be in the Department; 80-90% of its time is so spent. The 1108 is more heavily research oriented, perhaps 70-80% research. Why do you ask? (3) What? (4) The Diamondback did run a story on the end-of-semester crunch, but it wasn't noted for its illumination of the subject. The reporter(?) who called me, for example, had very little idea of what the computer was used for in classes, and seemed more interested in getting a statement that the Department hates the Center than in any information. Probably some parts of the Department do hate the Center (after all, we have to use the Univac systems), but the primary problem is with the State, and if it hurts to have to support more Univac, well, things could be worse. Ed.

We are a modern Department. Not only do our brochures feature a photograph (in tasteful dark blue) of the IBM 7090 console, MQ shamelessly visable, but I find that my far-out ALGOL 68 seminar for the fall has become a frumpy ALGOL 60 in the time schedule.

local buzz

GRAMMAR - an old-woman whose children have children.

PROPERTY GRAMMAR - an old woman with lots of land you might inherit.

REGULAR GRAMMAR - an old woman who is land poor.

POST MORTEM DUMP - property inherited from regular grammar.

TERMINAL VARIABLE - an intermittent storage-tube display.

NONTERMINAL VARIABLE - a Univac 1108.

SYNTAX - revenue-producing device levied against alcohol or excess page charge.

SEMANTICS - source of sailors' rocky mountain fever.

FIXED HEAD - a reformed junkie.

MOVABLE HEAD - a chemical toilet.

FLYING HEAD - a berserk chemical toilet.

DISPATCHER - a programmer who removes other programmers' fixes.

PROGRAM COUNTER - table where jobs are submitted.

NUMBER - softshoe routine intended to hide software problems from boss.

MCORE - request by boss for another number.

ER - mistake made by Univac executive, as in "to er is (in this case not) human..."

GENERATION GAP - 1108 - 7094 = -5986.

IBANK - source of cornea transplants.

DBANK - flatten out tilted highway curves.

--(mostly) G. Stockman

ALGEBRA DYNAMIC (more)

Students intrigued by Dr. Noonan's article on 'Dynamic Algebra' in the last PRINTOUT may also be interested in knowing that there are two language systems on the Univac 1100 machines in which the use of these techniques is of significant practical importance; they are APL and (believe it or not!) the assembler.

The Dynamic Algebra version of the infix-to-polish translator given as an example can be typed in and executed as an APL function almost exactly as it stands. Minor changes have been made: the array *SHS* represents the precedence of stacked operators, and the auxiliary variable *Q* has been defined for convenience. In addition, since *STP* can never be less than two, the expression (*ICP*≠0)∧(*ICP*>*STP*) has been reduced to (*ICP*>*STP*). The resulting APL function is:

```
  ∇ ALGEBRAIC ARG;POL;OPS;SO;ICP;STP;Q;I;J;K;SHR;SHS;ALFABET;OMEGA
[1]   INITIALIZE
[2]  WHILE:→1+ENDWHILE IF SO[I]=OMEGA
[3]   ICP←SHR[SO[I]]
[4]   STP←SHS[OPS[J]]
[5]   Q←((ICP=1)∧(STP=2))∨(ICP>STP)
[6]   POL[K]←((ICP=0)×SO[I])+(ICP≠0)×((Q=0)×OPS[J])
[7]   K←K+(Q=0)
[8]   J←J+(ICP≠0)×((ICP>STP)-(ICP≤STP))
[9]   OPS[J]←((ICP≤STP)×OPS[J])+(ICP>STP)×SO[I]
[10]  I←I+(ICP=0)∨Q
[11] ENDWHILE:→WHILE
[12]  CLEANUP
  ∇
```

A working version of this function complete with initialization may be found in public workspace 102 *DYNAMIC*. An unfortunate dearth of terminals with the APL characterset will make a listing of it painful to read, however.

Although the version of the example given above will work perfectly well, it does not make use of features of APL which are especially useful in applications of this kind. Notice that in the expression for *POL*[*K*] (line 6), the logical expressions in the variables *ICP* and *Q* are used to select one of the quantities *SO*[*I*] or *OPS*[*J*] to be stored into the output Polish string. APL has several functions designed for just this purpose and called, strangely enough, 'selection functions'. Of particular interest in this example is the selection function 'compress', represented by the character '/' (note: APL denotes division with a real divide sign, ÷). The syntax for compress is:

```
MASK/STRING
```

where *MASK* is a string of zeroes and ones, and *STRING* is a vector of characters or numbers (note: compression also works on higher-dimensional arrays such as matrices, but that's another story). Each bit in the mask is matched on a one-for-one basis with the corresponding elements in the right argument; if the mask bit is one, the matching element is selected. The result is a new string of elements selected by the ones in the mask. Notice that if the mask is all zeroes, nothing is selected and the result is a null string. Using the additional APL function catenate (represented by a comma), the expression for the output Polish string can be written as:

```
POL←POL,((ICP=0),(ICP≠0)∧(Q=0))/(SO[I],OPS[J])
```

In the above statement, *ICP*=0 will select *SO*[*I*], (*ICP*≠0)∧(*Q*=0) will select *OPS*[*J*], and if neither is true, nothing is selected. The (possibly null) result is then catenated onto the end of the output string, giving a new output string. The techniques of Dynamic Algebra are quite commonly used to form arguments for selection functions in APL. A version of the example using APL selection functions may also be found in workspace 102 *DYNAMIC*. This function (called *SELECTIVE*) also uses the APL selection function 'drop' (↓) to remove elements from strings. Since the strings are built dynamically with selection functions, the variables *I* , *J* , and *K* are not needed. This results in a further reduction in the length of the program to 7 lines (exclusive of initialization, etc.)

For those with a penchant for the kinky, given below is a version of the translator written in the assembler's language. Note that this is not 1100 series 'machine language', but is the internal meta-language of the assembler itself. The assembler's assignment statement is obviously EQU, its transfer statement is GO, and NAME is the statement which defines labels. The only conditional statement in the language is DO and it is used for both IF and WHILE constructions. It is this lack of control structures which makes algebraic techniques almost mandatory in writing assembler PROC's.

In the example below, 'noise' subscripts of the form (0) have been added to the variables ICP, STP, I, etc. This is necessary because the @ASM processor will give warning diagnostics (D-flags) if the value of a non-subscripted variable is modified.

```
P          PROC    *1
ALGEBRAIC* NAME
           INITIALIZE
WHILE      NAME
 DO SO(I(0))='#' , GO ENDWHILE
ICP(0)     EQU SHR(SO(I(0)))
STP(0)     EQU SHS(OPS(J(0)))
Q(0)       EQU ((ICP(0)=1)**(STP(0)=2))++(ICP(0)>STP(0))
POL(K(0))  EQU (ICP(0)=0)*SO(I(0))+(ICP(0)>0)*(Q(0)=0)*OPS(J(0))
K(0)       EQU K(0)+(Q(0)=0)
J(0)       EQU J(0)+(ICP(0)>0)*((ICP(0)>STP(0))-(ICP(0)<STP(0)+1))
OPS(J(0))  EQU (ICP(0)<STP(0)+1)*OPS(J(0))+(ICP(0)>STP(0))*SO(I(0))
I(0)       EQU I(0)+((ICP(0)=0)++Q(0))
 GO WHILE
ENDWHILE   NAME
           CLEANUP
           END
```

A working version of this routine complete with initialization and cleanup may be found in DYNAMIC*ALGEBRA.INFIX/POLISH . In examining it, you will note that there is both good and bad news. The good news is that character strings may be used as subscripts (as long as the string is no longer than 18 bits). The bad news is that the assembler has no PRINT statement. The results of the translation (the array POL) are displayed by the sample program in octal. It is left to the interested and/or perverted student to write his own 'instant compiler' by adding PROCs which will use the values in the array POL to generate machine instructions to actually evaluate the original infix expression.

-- P. E. Hagerty

Professional News

Dr. Charles J. Rieger, III has been accorded the honor of presenting the "Computers and Thought" lecture at the Fourth International Joint Conference on Artificial Intelligence at Ablisi, USSR, In September. The award is funded by proceeds from the book by Feigenbaum and Feldman, and recognizes outstanding contributions to the field of artificial intelligence.

Prof. William F. Atchison has been selected to receive the 1975 Chester Morrill Memorial Award presented by the Chesapeake Division of the Association for Systems Management. The award recognizes outstanding achievement in the area of education in the field of systems.

Dr. John Gannon holds a General Research Board award supporting his research this summer.

Ms. Patricia Merson has requested leave-without-pay indefinitely, starting April 25.

The $200 prize for outstanding undergraduate performance by a computer science graduate was awarded to Paul McMullin.

Prof. Robert G. Glasser will be returning to full time in the Physics Department in the fall.

Prof. Jack Minker chaired a session of the 1975 Conference on Information Science and Systems at Johns Hopkins in April. He delivered addresses on question-answering and problem-solving systems at NSA and Jet Propulsion Laboratory in April and May.

The National Science Foundation has awarded Prof. Yaohan Chu a grant for $104,000 for work on microprogramming.

Dr. Victor Basili delivered an invited talk on on the SIMPL family of languages and compilers at the University of Virginia, Charlottesville, in April.

Prof. Werner Rheinboldt presented an invited lecture "On the Solution of Sets of Nonlinear Equations Arising in the Applications of Finite Element Methods" at Brunel University, Uxbridge, Middlesex, England, in April.

Prof. Azriel Rosenfeld presented an invited lecture "Array and Web Languages: An Overview" at the Conference on Formal Languages, Automata, and Development, in Noordwijkerhout, the Netherlands.

Publications, Etc.

Harada, K. and M. V. Zelkowitz. インタラクティブ PL/I システムの設計と作成 Information Processing 16 (Feb., 1975), Information Processing Society of Japan, 85-92.

V. R. Basili and J. Knight. A language design for vector machines. SIGPLAN Notices, March, 1975.

V. R. Basili and A. Turner. SIMPL-T and the SIMPL family of Programming Languages. USE Technical Papers, Spring, 1975.

Carriage Control

I have just thought of a marvelously simple construction which shows that nothing can be computed in polynomial time by a nondeterministic Turing machine which could not as well have been done in polynomial time by a deterministic machine. Each nondeterministic machine represents a class of deterministic algorithms, and this class is effectively generated in the sense...Alas! the column is too short to contain it.