The only thing that never looks right is a rule. There is not in existence a page with a rule on it that cannot be instantly and obviously improved by taking that rule out.

George Bernard Shaw
The Dolphin 4 (1940), p. 81

TUGBOAT

THE TEX USERS GROUP NEWSLETTER
EDITOR ROBERT WELLAND

VOLUME 3, NUMBER 1 MARCH 1982
PROVIDENCE RHODE ISLAND U.S.A.

ADDRESSES OF OFFICERS, AUTHORS AND OTHERS

AIELLO, Luigia
Istituto di Informatica
Università degli Studi di Ancona
via della Montagnola 30
60100 Ancona, Italy

ARMADING, Tom
Medical Center Computing Facility
University of Rochester
Rochester, NY 14627
716-275-2613

BEEMAN, Roger L.
Boeing Aerospace Company
P O Box 39999
Mail Stop 86-13
Seattle, WA 98124
206-773-0770

BEETON, Barbara
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
401-272-9500

BENNETT, J. Michael
Regional EDP Ctr, Aarhus Univ
Ny Munkegade
8000 Aarhus C, Denmark

BERTELSEN, Erik
Regional EDP Center, Univ of Aarhus (RECAU)
Ny Munkegade
Bygning 540
DK-8000 Aarhus C, Denmark
45 6 128355; Telex: 64 754 recau dk

BLANFORD, Mark
Division 5531
Sandia National Laboratories
Albuquerque, NM 87185
505-844-8999

CANZII, Giovanni
Istituto di Cibernetica
Via Viotti 5
20133 Milano, Italy
02/235293

CARNES, Lance
163 Linden Lane
Mill Valley, CA 94941
415-388-8853

DOHERTY, Barry
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
401-272-9500

FRISCH, Michael J.
University Computing Center
208 Union St. S.E.
University of Minnesota
Minneapolis, MI 55455
612-373-4599

FUCHS, David
Department of Computer Science
Stanford University
Stanford, CA 94305
415-497-1646

GOUCHER, Raymond E.
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
401-272-9500

HOWERTON, Charles P.
National Bureau of Standards
Boulder Labs, room 3562
325 South Broadway
Boulder, CO 80303
303-499-1000 ext 4433

JACKSON, Calvin W.
Computer Science Department
California Inst of Tech
256-80
Pasadena, CA 91125
213-356-6245

KELLERMAN, David
Oregon Software
2340 SW Canyon Rd
Portland, OR 97201
503-226-7760

KNUTH, Donald E.
Department of Computer Science
Stanford University
Stanford, CA 94305

LUCARELLA, D.
ENEL, Automation Research Center
Milan, Italy

McCOURT, Scott
Burroughs Corporation
BCD Project
Corporate Drive, Commerce Park
Danbury, CT 06810
203-794-6191

McGAFFEY, Robert
Computer Sciences
Union Carbide, Nuclear Division
P O Box Y
Oak Ridge, TN 37830
615-574-5445

MORRIS, Robert
Mathematics Department
UMASS at Boston
Boston, MA 02125
617-287-1900, ext. 2545

NICHOLS, Monte C.
Exploratory Chemistry Division
Sandia National Laboratories 8313
Livermore, CA 94550
415-422-2906

PALAIS, Richard S.
Department of Mathematics
Brandeis University
Waltham, MA 02154
(On leave through June 1982)

PAVAN, S.
University of Pisa
ISI, Corso Italia 40
I-56100 Pisa, Italy

PENNY, S. Keith
Computer Sciences
Union Carbide, Nuclear Division
P O Box X
Oak Ridge, TN 37830

PILENGA, A.
Institute of Cybernetics
University of Milan
Via Viotti 5
20133 Milan, Italy

PIZER, Arnold
Department of Mathematics
University of Rochester
Rochester, NY 14627
716-275-4428

PLASS, Michael F.
Imaging Sciences Laboratory
Xerox Corporation
3333 Coyote Hill Road
Palo Alto, CA 94304
415-494-4810

PLASS, Susan
Polya 203
Center for Information Technology
Stanford University
Stanford, CA 94305
415-497-1322

PLATT, Craig R.
Dept of Math, Machray Hall
Univ of Manitoba
Winnipeg (R3T 2N2), Manitoba, Canada
204-474-9832

PRICE, Lynne A.
CALMA
Research and Development
212 Gibraltar Drive
Sunnyvale, CA 94086
408-744-1950

SAMUEL, Arthur L.
Computer Science Department
Stanford University
Margaret Jacks Hall 436A
Stanford, CA 94305
415-497-3330

SPIVAK, Michael
2478 Woodridge Drive
Decatur, GA 30033
404-329-0372

STROMQUIST, Ralph
MACC
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706
608-262-8821

TRABB-PARDO, Luis
Department of Computer Science
Stanford University
Stanford, CA 94305

WELLAND, Robert
Department of Mathematics
Northwestern University
2033 Sheridan Road
Evanston, IL 60201
312-864-2898

WENDEL, Whitney
Addison-Wesley
Jacob Way
Reading, MA 01867
617-944-3700 x2304

WHIDDEN, Samuel B.
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
401-272-9500

WHITNEY, Ronald
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
401-272-9500

ZABALA, Ignacio
Department of Computer Science
Stanford University
Stanford, CA 94305

ZAPF, Hermann
Seitersweg 35
D-6100 Darmstadt Fed Rep Germany

ZIPPEL, Richard
Massachusetts Institute of Technology
545 Tech Square
Cambridge, MA 01239

TUGboat, the newsletter of the TEX Users Group (TUG), is published irregularly for TUG by the American Mathematical Society, P.O. Box 6248, Providence, RI 02940. Annual dues for individual members of TUG, $15.00 for 1982, include one subscription to TUGboat. Applications for membership in TUG should be addressed to the TEX Users Group, c/o American Mathematical Society, P.O. Box 1571, Annex Station, Providence, RI 02901; applications must be accompanied by payment.

Manuscripts should be submitted to a member of the TUGboat Editorial Committee, whose names and addresses are listed inside the front cover. Articles of general interest, or not covered by any of the topics listed, should be sent to Robert Welland, Editor-in-Chief, at the address shown. Items submitted on magnetic tape should be addressed to Barbara Beeton, American Mathematical Society, P.O. Box 6248, Providence, RI 02940.

Submissions to TUGboat are for the most part reproduced with minimal editing. Any questions regarding the content or accuracy of particular items should be directed to the authors.

OFFICIAL ANNOUNCEMENTS

TUG Meeting, July 26–27, 1982, Stanford, California

A meeting of the TEX Users Group will be held at Stanford University on Monday and Tuesday, July 26–27, 1982. TEX82 will be introduced at the meeting, and the Stanford group plans to conduct demonstrations on several different architectures. A preliminary program, housing and registration information will be mailed out in mid-May.

Individual Membership Dues and Privileges

1982 dues for individual members of TUG are $15. Membership privileges include all issues of TUGboat published during the membership (calendar) year. All new members and other persons inquiring about TUG will be sent a complimentary copy of TUGboat Vol. 1, No. 1 (1980). Membership for 1981 (Volume 2) will be accepted through April 30, 1982, at $10, but thereafter the price for back issues will be $10 each. Members residing outside North America may, on payment of a supplementary fee of $12 per subscription or volume year, have TUGboat air mailed to them. (Members who wish to add this option for the remaining 1982 issues may do so by paying a supplementary fee of $7 per subscription.) Lengthy macro packages, such as Max Díaz's *Fácil TEX* (Appendix A, TUGboat Vol. 2, No. 2), will be published separately in the future; details will be given on the order form.

TUGboat Schedule

The deadline for submitting items for Vol. 3, No. 2, will be September 1, 1982; the mailing date will be October 1. Contributions on magnetic tape or in camera copy form are encouraged; see the statement of editorial policy below. Editorial addresses are given on the inside front cover, and a form containing instructions for submitting items on tape is bound into the back of this issue.

It is TUG's policy to keep all issues of TUGboat in print. Each member is entitled to receive all issues which appear during the membership year, as well as Vol. 1, No. 1. Domestic subscriptions are mailed third class bulk, which may take up to six weeks to reach its destination; foreign shipments are surface printed matter, unless the air mail option is elected. If you have not received an issue to which you are entitled, write to TUG at the address given on the order form for general correspondence.

* * * * * * * * * * *

General Delivery

* * * * * * * * * * *

STATEMENT OF EDITORIAL POLICY

Barbara Beeton

Since it was launched, TUGboat has been published for TUG by the American Mathematical Society. Through 1981, the editorial and production time necessary to ready copy for the printer was provided gratis. The Society can no longer provide these services at no charge, and beginning with the present issue, time devoted to that task will incur charges at rates the same as those for internal Society users of similar services. In addition, my time has become less available, although I will continue to retain control over the final quality assurance inspection of each issue. I will no longer be able to extend reminders to potential contributors— if authors don't remember to prepare and submit articles, TUGboat issues may be slim.

Some decisions made at the Cincinnati TUG meeting will result in changes in TUGboat editorial policy, the chief intention being to decrease the amount of time required to prepare copy for the printer. One change is the distribution of effort among Associate Editors. Lynne Price has been editing the Macro and Problems columns for several issues, and she is now joined by other volunteers, whose names and addresses are listed inside the front cover. If you are writing an article in one of the areas listed, please submit paper copy to the appropriate editor; articles of general interest, or in areas not listed, should go to Editor-in-Chief Bob Welland. Tapes are still welcome, and can be sent directly to me. (See Vol. 2: No. 1, page 53, and No. 3, page 23, and the form in the back of this issue for details on tape content and format.) It is not intended that all columns appear in all issues: if there is no traffic in a particular area, there will be no column. On the other hand, if traffic is exceptionally heavy in a particular area at any point, consideration will be given to publishing a "topical" issue.

It was suggested in Cincinnati that issues be published less frequently. In 1982, an issue will be published after every general meeting in order to report to the membership what happened. The deadline for manuscripts will be a month to six weeks after the end of the meeting. In between, any manuscripts received in Providence will be held until the next scheduled issue, unless it becomes obvious that enough material exists, or an associate editor volunteers to take charge of a special issue.

Copy is solicited in camera copy form, when possible. If copy has been prepared by TEX and is legible, it will be used as submitted, reduced photographically if necessary (which is advisable for copy prepared on an output device with 200 dot/inch or lower resolution), with running heads applied. The dimensions used in the TUGboat header files are: `\vsize 54pc`, for one-column pages `\hsize 39pc`, and for two-column pages `\hsize 18.75pc` and `\pagewd 39pc`. If the copy is to contain headers which should not be covered up by the TUGboat running heads, 54pc should be used as the length of the full page. The type used for ordinary text is `cmr10`, on `\baselineskip 12pt`.

Deadlines will be firmly adhered to. Any material received in Providence later than the published deadline (in the announcements box of every issue) will be consigned to the back of the book, as "Late-Breaking News", or else held over for the next issue.

Since TUGboat is itself an advertisement for TEX, it is not our intention to lower quality, but to streamline production. Your attention to formatting of material submitted as camera copy and to the content and commands in material submitted on tape will assist greatly in reaching that goal.

* * * * * * * * * * * *

REPORT ON THE TUG STEERING COMMITTEE MEETING

The Steering Committee meeting in Cincinnati took place in several sessions. At the first, on January 11, the role of the American Mathematical Society in future production of TUGboat was discussed, and other items were suggested for discussion at the second session, an open meeting on January 12.

The following actions were taken, either by the Steering Committee alone or at the open meeting:

a. Membership for 1981 will be available retroactively through April 30, at $10.00; thereafter TUGboat Volume 2 will be available at the price of $10.00 per back issue.

b. Ordinary subscriptions will be accepted for TUGboat at the same price as individual membership; this is intended primarily for the convenience of libraries.

c. Effective with the first 1982 issue of TUGboat, the American Mathematical Society can no longer provide free editorial and production services; these services will be charged to TUG at the same rates incurred by internal Society users of similar services. Other actions will be taken to streamline production while maintaining satisfactory quality; see the Statement of Editorial Policy by Barbara Beeton (page 3) for details.

d. A rough budget was drawn up and presented to the membership, showing the expected cost of various TUG functions for 1982. A redrafted version appears on page 45.

e. Steering Committee members will be permitted to attend TUG workshops at no charge if they are unable to obtain support from their institutions.

f. The Finance Committee was requested to investigate the sale of mailing lists and advertisements in TUGboat, after soliciting opinions on the legal and tax consequences of such sales. They were also requested to obtain opinions on the legal and tax consequences of receiving fees for membership, subscriptions and royalties.

g. The price of Don Knuth's manual for TEX82 will be increased by $1.00, which will be paid as a royalty to TUG.

h. A Bylaws Committee was appointed, consisting of Bob Morris, Susan Plass, Lance Carnes, Dave Kellerman, and Craig Platt. They will prepare a report for the next meeting.

i. Institutional membership will be instituted when TEX82 is ready for distribution. Dues of

$250 for not-for-profit, and $500 for profit making organizations were suggested. A 250/500 Committee was appointed, consisting of Luis Trabb-Pardo, Arnie Pizer, Mark Blanford, Calvin Jackson and Whit Wendel, to determine the details of membership privileges, and held an informal meeting. Suggested benefits: granting of a "license" to obtain TeX82, either the "vanilla" version or a custom-tailored version through the appropriate distributor (this would not preclude a distributor from charging a fee for tape creation or recovery of expenses); a specific number (perhaps 5) copies of TUGboat; an annual tape of contributed macros.

j. Bob Morris reported for the Tape Committee that criticism had been solicited regarding the proposal for an interchange standard (Vol. 2 No. 2, page 10). The committee will adopt or revise the proposal to be presented at the next meeting. The standard should accommodate font files as well as program and data files.

k. The next meeting will be held at Stanford on Monday and Tuesday, July 26–27, immediately following a SIAM meeting. TeX82 will be introduced at the meeting, and participants will be able to see it in operation on a number of different architectures. Susan Plass volunteered to be in charge of local arrangements.

Minutes respectfully submitted,

Robert A. Morris
Secretary

Editor's note: Attendees at the meeting may submit additions and corrections to the minutes in writing to the Secretary.

* * * * * * * * * * *

Program, TUG Winter Meeting
Cincinnati, January 11–12, 1982

Don Knuth – WEB systems of structured documentation
Don Knuth – the TeX82 system
David Fuchs – interfaces and photo-typesetters
Steering Committee open meeting
Michael Plass – on font design
Lance Carnes – TeX on the HP-3000
Jack Schneble – TeX at McGraw-Hill
Ron Whitney – TeX at AMS
Michael Spivak – the AMS-TeX macro package
Monte Nichols – chemical symbol macros
Lynne Price – macro panel discussion
Site Coordinators – individual meetings:
CDC Cyber (Erik Bertelsen)
DEC 10/DEC 20 (Barry Doherty and Arnie Pizer)
IBM 370 (Susan Plass)
small architectures (Lance Carnes)
VAX/UNIX (Bob Morris)
VAX/VMS (Monte Nichols)

* * * * * * * * * * *

Attendees, TUG Winter Meeting
Cincinnati, January 11–12, 1982

Bechtel, Brian – Bell-Northern Research, Inc.
Beeman, Roger L. – Boeing Aerospace Corp.
Beeton, Barbara – American Mathematical Society
Bertelsen, Erik – University of Aarhus
Blanford, Mark L. – Sandia National Laboratory
Burgart, Calvin E. – Science Applications, Inc.
Carnes, Lance – Gentry, Inc.
Chaletzky, Kenneth B. – Circle Graphics, Inc.
Collins, Herman – University of Kentucky
Crawford, John – Ohio State University
Crumly, James D. – Hewlett-Packard Co.
Doherty, Barry C. W. – American Mathematical Society
Dupree, Charles – Digital Equipment Corp.
Eisenbraun, James – Eisenbraun Co.
Fuchs, David – Stanford University
Gabelnick, Stephen – Argonne National Laboratory
Greenberg, Lewis H. – Michigan State University
Grosso, Paul – University of Michigan
Hagins, William A. – National Institute of Health
Hauck, Roger – Smithsonian Astrophysical Observatory
Hetzel, Allan – University of Kentucky
Hickey, Thomas B. – Online Computer Library, Inc.
Horn, Roger A. – Johns Hopkins University
Ion, Patrick D. – Mathematical Reviews/AMS
Jackson, Calvin W. – California Insititute of Technology
Janson, Barbara – American Mathematical Society
Kellerman, David – Oregon Software
Kelly, William H. – University of Wisconsin
Kerekes, Alan S. – Union Carbide Nuclear Division
Knuth, Donald E. – Stanford University
Kocian, Raymond – Schlumberger-Doll Research Center
Krohm, Gary W. – Krohm International
Link, John P. – George Washington University
Macewich, Michael – Quaker Oats
Maloney, Linda S. – Shepard's/McGraw-Hill
Mase, Georgia – Online Computer Library, Inc.
McCall, Thomas – Colorado State University
McGaffey, Robert W. – Union Carbide Nuclear Division
Mooney, James D. – University of West Virginia
Morris, Robert – University of Massachusetts at Boston
Nichols, Monte C. – Sandia National Laboratory
Palmer, Theodore W. – University of Oregon
Penny, S. Keith – Union Carbide Nuclear Division
Perkins, Maria T. – Vanderbilt University
Pierce, Thomas H. – Rohm & Haas Research Laboratory
Pizer, Arnold – University of Rochester
Plass, Michael – Xerox Corp.
Plass, Susan – Stanford University
Platt, Craig R. – University of Manitoba
Price, Lynne A. – CALMA
Pritchard, Paul
Rabinowitz, Stanley
Rhoads, Forrest D. – Shepard's/McGraw-Hill
Roberts, Carol – Mathematical Reviews/AMS
Rodgers, David – University of Michigan
Rossi, Michael J. – Grumman Aerospace Corp.
Rowland, Joseph – Metromail
Samuel, Arthur – Stanford University
Schneble, Jack – McGraw-Hill, Inc.
Smith, George – Ohio State University
Spivak, Michael
Standefer, Lee – Metromail
Stork, Carl – Microsoft
Thacher, Clarke – University of Kentucky
Thedford, Rilla – Mathematical Reviews/AMS
Trabb-Pardo, Luis – Stanford University
Vogt, Joann B. – Vanderbilt University
Walsh, Robert J. – Harvard University
Wendel, Whitney – Addison-Wesley
Whidden, Samuel B. – American Mathematical Society
Whitney, Ronald F. – American Mathematical Society
Widmayer, Peter – University of Karlsruhe
Woolf, William B. – Mathematical Reviews/AMS
Zabala, Ignacio – Stanford University
Zwick, Dan – University of Bonn

* * * * * * * * * * *

Editor's note: The following letter from Robert McGaffey and Keith Penny of Union Carbide responds to some questions raised at the Cincinnati meeting concerning how to keep TUG a viable organization, to foster the use and growth of TeX. Some of the ideas have already been approved (see Bob Morris' report, page 4). Comments and suggestions can be sent to any member of the Steering Committee or submitted for publication in the next issue.

February 3, 1982

An open letter to TUG:

This letter expresses some of our thoughts on the organisation and work of TUG. We wish to have some input into the operation of TUG and so have organised our thoughts to present them to both the Steering Committee and TUG. In what follows, TEX System is intended to include TEX , Metafont, TEX macros, and device drivers needed to have a working system.

Goals: TEX System users want to have the best typesetter available for their work. In order to accomplish this TUG wants the TEX System to continue to develop in a direction that improves the overall typesetting system. Obviously, the more users the TEX System has the more suggestions for improvement and free development TUG will have. So TUG should expand as much as possible. The fact that the TEX System for the most part is in the public domain coupled with the free development the TEX System receives means that before long there will be many versions of TEX and many output device drivers and many macro packages some of which will be excellent and some which will in fact detract from the TEX System. To prevent this problem TUG needs some means of assuring quality control.

The above goals can perhaps be accomplished if TUG operates as a business with the goal of setting up the TEX System on as many computers and output devices as possible. Since the TEX System cannot be sold, TUG could operate as though it is trying to make money selling TEX and macro manuals.

Income: To finance TUG's operation the following four suggestions are made (at least two of which have been decided on already).

1. Offer an institutional membership for 250 or 500 dollars. Income from these memberships should go into the general treasury. If many institutions are willing to pay for up-to-date copies of TEX and the macro package then this will be a good source of income. "If" is used because of the fear that many institutions will be content to have free copies of the TEX System since they can legally copy it because it is in the public domain.

2. Offer subscriptions to TUGBOAT (not memberships to TUG) which pay for the entire cost of printing and mailing issues of TUGBOAT. Keep this money separate from the general fund. The advantage of this policy is that TUG doesn't have to worry about losing money through financing TUGBOAT. A subscription to TUGBOAT should allow the subscriber to be a member of TUG. Wording the offer in this way saves red tape for many who become subscribers through their own institutions.

3. Offer advertising in TUGBOAT and use the revenue to keep the costs

of TUGBOAT at a reasonable amount. Add the extra money from the sale of advertisements to the general fund.

4. Add an additional dollar, or even five dollars, to the cost of each TeX and macro manual. This revenue should go into the general fund.

Organization: TUG should be organized as a business. Creation of the following groups with their activities is suggested:

1. Officers—elected by TUG including President, Vice President, Secretary, Treasurer, hopefully aware that TUG should be business oriented. (In keeping with good punmenship these should be named the Skipper, the XO, the Yeoman, and the Purser.)

2. Management Committee—reports to both the officers and to the Steering Committee. Responsible for seeing that TUG's operations are handled efficiently. All other committees report to the Management Committee. It is even possible for TUG to retain or hire a company to handle all business transactions. SHARE, the IBM user's group, does this.

3. Individuals hired on an as needed basis by the Management Committee or volunteers to handle certain aspects of management. For example, lawyers, CPA's salesmen for TUGBOAT advertisements, those who organize meetings, etc.

4. TUGBOAT Committee—those responsible for both printing and mailing TUGBOAT.

5. Steering Committee—reports to officers and is on equal footing with the Management Committee. This committee decides the direction TeX System should take as far as technical development goes. For example, we would like to see TeX developed for micro computers. It would be nice if an interactive TeX (which actually showed the results as commands were entered) were developed. The Steering Committee also maintains a set of codes for distribution which reflect the best TeX available for each CPU as well as the best output device driver codes for each output setup. This assures TUG members that they are using the best codes for their installation. A TeX macro library should be maintained and a standard set of macros for distribution maintained.

6. TeX site coordinators—report to Steering Committee. These coordinators are responsible for maintaining the best Pascal version of TeX for the particular computer they have. Such coordinators should be able to instruct potential users in how to implement TeX on their CPU. These coordinators should not be charged with distribution of tapes.

7. Output device coordinators—report to Steering Committee. These coordinators are responsible for maintaining the best programs used to drive an output device. For example, there is the PDP-10–Z80–Versatec output device. There could be an IBM370–FR80 output device, etc. These coordinators should be able to furnish advice to users with the same hardware output configuration. These coordinators should not be charged with distribution of tapes. Note that

coordinators are charged with keeping a copy of their programs available to the Steering Committee along with algorithm and implementation documentation. The reason for not requiring coordinators to distribute tapes is that many institutions do not allow their employees to do such work and thus if TUG requires its coordinators to distribute codes TUG will not have the best software available for each hardware configuration.

8. Distribution Committee—reports to Steering Committee. Responsible for distributing manuals to users and tapes to responsible individuals at given institutions. (The committee chairman is the Quartermaster.)

Expenditures: First, let it be pointed out that the large majority of TUG members could care less how the money is spent because it isn't their money. As a result of this there is little reason to have the general membership vote on questions of finance. Five areas for spending money are suggested:

1. Hire and/or retain the professional people TUG needs such as a lawyer, a CPA, etc. Part-time basis should serve for most of these positions.

2. Pay someone to coordinate the development of macro packages so that a good "official" set of macros can be distributed.

3. Pay the expenses of the officers, members of the Steering Committee and Management Committee incurred as a result of travel to TUG meetings.

4. Pay selected TUG volunteers to demonstrate and perhaps even implement TeX at prospective institutions. Obviously, the Steering Committee would need to exercise much care in selecting TUG members to do this.

5. Pay selected volunteers to develop macro packages and make improvements to TeX itself.

Thank you for your attention.

Sincerely yours,

Robert McGaffey Keith Penny

Robert McGaffey
Keith Penny

Computer Sciences
Union Carbide, Nuclear Division

REPORT ON THE ANSI X3J6 MEETING

Lynne A. Price

Supported by TUG, I spent January 25–29 in Lancaster, Pennsylvania attending a meeting of the ANSI X3J6 Text Processing Language Standards Committee. The committee is defining a standard language with facilities for text editing, text formatting, and generalized markup. For text editing, the object is to identify basic editing functions and a macro facility so that an individual user can take a personal macro file from system to system and not need to learn a new editor for each computer. For text formatting, the object is to be able to produce readable output on different systems from a single source file. It is understood that line breaks, hyphenation, page breaks, and so on cannot be preserved across different facilities. The output devices considered include daisy-wheel printers, word-processing equipment, and high-resolution typesetters. Text markup refers to labelling elements of a document—titles, chapters, footnotes, etc. The goal for generalized markup is to itemize the elements needed for common types of documents, so that input for various document formatters could be automatically prepared from a source file containing the text to be formatted interspersed with markup codes. Thus, preprocessors might exist to translate source files from the standard markup language to TEX input form, to SCRIBE input form, to APS-5 input form, etc.

Of the fifteen individuals in Lancaster, approximately half were committee members (to join, an individual must attend two meetings and pay $100). This attendance is fairly typical, although the mailing list has about sixty names. The committee has been meeting four times a year, for week-long sessions. The next meetings are scheduled for Phoenix in April, Edmonton in August, New Hampshire in October, and the Bay Area in January or February. If the current schedule, which calls for completion of the standard in 1983, can be met, only three meetings will be required next year. Once the standard is approved, the committee will continue to have short meetings once or twice a year; activity will then increase as the five-year review approaches.

I can forward a copy of the not-yet-completed draft standard (dated just before the Lancaster meeting) to any interested TUG member. The X3J6 formatting language has been greatly influenced by the concepts of boxes and glue as used by TEX. It is currently assumed that it will be easy to translate, in both directions, between the eventual standard language and TEX. Several committee members also belong to TUG. However, none of the X3J6 members in Lancaster yet has access to TEX. As a TEX user, I was repeatedly able to contribute to the discussion. During the week, topics pertinent to formatting ranged over paragraph justification, word spacing, letter spacing, line spacing, leaders, rules, and page layout. I learned quite a bit about typesetting. Subtopics I found most interesting involved generalizations of structures and algorithms used by TEX.

It is very clear that X3J6 can benefit from involvement by TUG. There are advantages to the TEX community as well. X3J6 is formed of individuals knowledgeable in both typesetting and automatic text processing. Until the TEX language stabilizes, X3J6 can comment on its applicability to general, non-mathematical typesetting. There has always been interest within TUG in a possible "Son of TEX"; X3J6 may be an outlet for future generalizations. Finally, X3J6 and TUG have a common interest in separating font sales from sales of typesetting equipment. For the above reasons, I recommend that TUG continue to finance a representative at X3J6 meetings. Although we granted the Finance Committee authority to make this decision in Cincinnati, we can all provide input to the process through TUGboat, mail, and telephone.

* * * * * * * * * * *

Software

* * * * * * * * * * *

FIXED-POINT GLUE SETTING AN EXAMPLE OF WEB

Donald E. Knuth
Stanford University

The "definitive" version of TEX is being written in a new language called WEB, which is a mixture of TEX and PASCAL. I will soon be publishing a complete manual about WEB, but in the meantime I think it will be useful to have an example of a fairly short piece of code written in "web" form. Therefore I have prepared the accompanying program, which also serves another function: It illustrates how to remove the last vestiges of floating-point arithmetic from the new TEX.

The eleven pages that follow this introduction contain the example program in its "woven" form, including the table of contents and the two indices that are generated automatically. I hope the reader can guess how WEB works just by looking at this particular example. The PASCAL version of the TEX process or will eventually appear in the same format, only it will be somewhat longer.

The twelfth page, which is page 23 of this issue of TUGboat, is an example of the output generated by the fixed-point routines. And the page after that is the actual PASCAL program that was produced from the "web". (This PASCAL code isn't very readable, but it is intended to be read only by the PASCAL compiler, except in rare emergencies. It does contain cross-references that show where each numbered part of the web has been inserted.)

Following the PASCAL code I have attached an example page of the WEB file, which is what I actually typed into the computer. This file, GLUE.WEB, was the source of everything else. A program called TANGLE took GLUE.WEB as input and produced the PASCAL code GLUE.PAS as output; I never looked at that output, I just let PASCAL compile it. Another program called WEAVE took GLUE.WEB as input and produced GLUE.TEX as output. (A sample page of GLUE.TEX appears after the sample page of GLUE.WEB, so that you can see what WEAVE does.) When TeX processed GLUE.TEX, the result was the eleven pages that I mentioned first; you should read these eleven pages first.

How much computer time did this all take? I didn't gather exact data, which is not easy to obtain on our time-shared DEC-10 computer, but the following approximate times are fairly accurate: TANGLE took two seconds to convert the WEB file to the PAS file, PASCAL took two seconds to convert that to a REL file, the system loader took two seconds to get the program in memory, and the program produced its output in a small fraction of a second. Furthermore WEAVE took four seconds to convert the WEB file to the TEX file, TeX took 40 seconds to convert that to an output file (in this case a PRESS file for the Dover printer), and the hardcopy output was printed by the time I walked down one flight of stairs to the printer room. You have to multiply the TANGLE-PASCAL-load-run time by about 5, since I went through five passes while debugging; and you have to multiply the WEAVE-TeX-print time by 2, since this is my second draft.

How much human time did it take? I spent a full day considering various ways to do the necessary fixed-point computations, until deciding that this scheme was preferable to another that was based on two 16-bit integers instead of powers of 2. I spent about three hours writing the WEB code, about two hours typing it into the computer and editing it as I went, and about two hours proofreading and debugging.

The bugs turned out to be mostly typographical or related to fussy details, since the web structure made my program so clear (to me at least) that I was pretty sure it was correct as I wrote it. Here are the bugs I remember making:

1) I forgot that WEB doesn't allow me to use its special notation for octal constants in a comment, unless the constant appears in "PASCAL mode".
2) In one place I typed 'global' instead of 'Global', so WEB could not match the two names.
3) I left a dollar sign off at the end of a formula. (This later caused TeX to give an error message that I had an extra right brace; then it said I couldn't do something-or-other in restricted horizontal mode.)
4) I forgot that PASCAL doesn't allow a function to return a structured type.
5) I forgot to declare the variables a, b, and c in one procedure.
6) I used '*write*' instead of '*writeln*' in one place.
7) I left off the **begin** and **end** that now surround the module called ⟨Compute c by long division⟩.
8) I used s instead of ss in the so-called "easy case".

Note that there are bugs in my use of WEB, in my use of PASCAL, in my use of TeX, and in my algorithm. But I believe the total number of bugs would have been a lot more if I had programmed separately in PASCAL and written a separate description in TeX. And the final documentation is not only better than I know how to make by any other method, it also is guaranteed to be a documentation of exactly the program as it describes, since the documentation and the program were generated by the same WEB source file.

As I gain more experience with WEB, I am finding that it significantly improves my ability to write reliable programs quickly. This is a pleasant surprise, since I had designed WEB mainly as a documentation tool.

Fixed-point Glue Setting

	Section	Page
Introduction	1	2
The problem and a solution	4	3
Glue multiplication	8	5
Glue setting	12	6
Glue-set printing	15	7
The driver program	20	8
Index	27	10

This research was supported in part by the National Science Foundation under grants IST-7921977 and MCS-7723738; by Office of Naval Research grant N00014-81-K-0330; and by the IBM Corporation.

1. Introduction. If TEX is being implemented on a microcomputer that does 32-bit addition and subtraction, but with multiplication and division restricted to 16-bit multipliers and divisors, it can still do the computations associated with the setting of glue in a suitable way. This program illustrates one solution to the problem.

Another purpose of this program is to provide the first "short" example of the use of WEB.

2. The program itself is written in standard PASCAL. It begins with a normal program header, most of which will be filled in with other parts of this "web" as we are ready to introduce them.

```
program GLUE(input, output);
  type ⟨Types in the outer block 6⟩
  var ⟨Globals in the outer block 8⟩
  procedure initialize;  {this procedure gets things started}
    var ⟨Local variables for initialization 9⟩
    begin ⟨Set initial values 10⟩
    end;
```

3. Here are two macros for common programming idioms.

```
  define incr(#) ≡ # ← # + 1   {increase a variable by unity}
  define decr(#) ≡ # ← # − 1   {decrease a variable by unity}
```

4. The problem and a solution. We are concerned here with the "setting of glue" that occurs when a TEX box is being packaged. Let $x_1, \ldots, x_n$ be integers whose sum $s = x_1 + \cdots + x_n$ is positive, and let t be another positive integer. These x_i represent scaled amounts of glue in units of spt (scaled points), where one spt is 2^{-16} of a printer's point. The other quantity t represents the total by which the glue should stretch or shrink. Following the conventions of TEX82, we will assume that the integers we deal with are less than 2^{31} in absolute value.

After the glue has been set, the actual amounts of incremental glue space (in spt) will be the integers $f(x_1), \ldots, f(x_n)$, where f is a function that we wish to compute. We want $f(x)$ to be nearly proportional to x, and we also want the sum $f(x_1) + \cdots + f(x_n)$ to be nearly equal to t. If we were using floating-point arithmetic, we would simply compute $f(x) = (t/s) \cdot x$ and hope for the best; but the goal here is to compute a suitable f using only the fixed-point arithmetic operations of a typical "16-bit microcomputer."

The solution adopted here is to determine integers a, b, c such that

$$f(x) = \lfloor 2^{-b} c \lfloor 2^{-a} x \rfloor \rfloor$$

if x is positive. Thus, we take x and shift it right by a bits, then multiply by c (which is 2^{15} or less), and shift the product right by b bits. The quantities a, b, and c are to be chosen so that this calculation doesn't cause overflow and so that $f(x_1) + \cdots + f(x_n)$ is reasonably close to t.

The following method is used to calculate a and b: Suppose

$$y = \max_{1 \le i \le n} |x_i|.$$

Let d and e be the smallest integers such that $t < 2^d s$ and $y < 2^e$. Since s and t are less than 2^{31}, we hve $-30 \le d \le 31$ and $1 \le e \le 31$. An error message is given if $d + e > 31$; in such a case some x_m has $|x_m| \ge 2^{e-1}$ and we are trying to change $|x_m|$ to $|(t/s)x_m| \ge 2^{d+e-2} \ge 2^{30}$ spt, which TEX does not permit. (Consider, for example, the "worst case" situation $x_1 = 2^{30} + 1$, $x_2 = -2^{30}$, $t = 2^{31} - 1$; surely we need not bother trying to accommodate such anomalous combinations of values.) On the other hand if $d + e \le 31$, we set $a = e - 16$ and $b = 31 - d - e$. Notice that this choice of a guarantees that $\lfloor 2^{-a} x \rfloor < 2^{16}$. We will choose c to be at most 2^{15}, so that the product will be less than 2^{31}.

The computation of c is the tricky part. The "ideal" value for c would be $\rho = 2^{a+b} t/s$, since $f(x)$ should be approximately $(t/s) \cdot x$. Furthermore it is better to have c slightly larger than ρ, instead of slightly smaller, since the other operations in $f(x)$ have a downward bias. Therefore we shall compute $c = \lceil \rho \rceil$. Since $2^{a+b} t/s < 2^{a+b+d} = 2^{15}$, we have $c \le 2^{15}$ as desired.

We want to compute $c = \lceil \rho \rceil$ exactly in all cases. There is no difficulty if $s < 2^{15}$, since c can be computed directly using the formula $c = \lfloor (2^{a+b} t + s - 1)/s \rfloor$; we have $2^{a+b} t < 2^{15} s < 2^{30}$.

Otherwise let $s = s_1 2^l + s_0$, where $2^{14} \le s_1 < 2^{15}$ and $0 \le s_0 < 2^l$. We will essentially carry out a long division. Let t be "normalized" so that $2^{30} \le 2^h t < 2^{31}$ for some h. Then we form the quotient and remainder of $2^h t$ divided by s_1,

$$2^h t = q s_1 + r.$$

It follows that $2^{h+l} t - qs = 2^l r - s_0 q = R$, say. If $0 \ge R > -s$ we have $q = \lceil 2^{h+l} t/s \rceil$; otherwise we can replace (q, R) by $(q \pm 1, R \mp s)$ until R is in the correct range. It is not difficult to prove that q needs to be increased at most once and decreased at most seven times, since $2^l r - s_0 q < 2^l s_1 \le s$ and $s_0 q/s \le (2^h t/s_1)(s_0/2^l s_1) < 2^{31}/s_1^2 \le 8$. Finally $c = \lceil 2^{a+b-h-l} q \rceil$; and we have $a + b - h - l = -1$ or -2, since $2^{28+l} \le 2^{14} s = 2^{a+b+d-1} s \le 2^{a+b} t < 2^{a+b+d} s = 2^{15} s < 2^{30+l}$ and $2^{30} \le 2^h t < 2^{31}$.

An error analysis shows that these values of a, b, and c work satisfactorily, except in unusual cases where we wouldn't expect them to. We have

$$\begin{aligned} f(x) &= 2^{-b}(2^{a+b} t/s + \theta_0)(2^{-a} x - \theta_1) - \theta_2 \\ &= (t/s)x + \theta_0 2^{-a-b} x - \theta_1 2^a t/s - 2^{-b} \theta_0 \theta_1 - \theta_2 \end{aligned}$$

where $0 \le \theta_0, \theta_1, \theta_2 < 1$. Now $0 \le \theta_0 2^{-a-b} x < 2^{e-a-b} = 2^{d+e-15}$ and $0 \le \theta_1 2^a t/s < 2^{a+d} = 2^{d+e-16}$, and the other two terms are negligible. Therefore $f(x_1) + \cdots + f(x_n)$ differs from t by at most about $2^{d+e-15} n$. Since 2^{d+e} spt is larger than the largest stretching or shrinking of glue after expansion, the error is at worst about $n/32000$ times as much as this, so it is quite reasonable. For example, even if fill glue is being used to stretch 20 inches, the error will still be less than $\frac{1}{1600}$ of an inch.

5. To sum up: Given the positive integers s, t, and y as above, we set $a \leftarrow \lfloor \lg y \rfloor - 15$, $b \leftarrow 29 - \lfloor \lg y \rfloor - \lfloor \lg t/s \rfloor$, and $c \leftarrow \lceil 2^{a+b} t/s \rceil$. The implementation below shows how to do the job in PASCAL without using large numbers.

6. TEX wants to have the glue-setting information in a 32-bit data type called *glue_ratio*. The PASCAL implementation of TEX82 has *glue_ratio* = *real*, but alternative definitions of *glue_ratio* are explicitly allowed.

For our purposes we shall let *glue_ratio* be a record that is packed with three fields: The *a_part* will hold the positive integer $a + 16$, the *b_part* will hold the nonnegative integer b, and the *c_part* will hold the nonnegative integer c. Note that we have only about 25 bits of information in all, so it should fit in 32 bits with ease.

```
⟨Types in the outer block 6⟩ ≡
  glue_ratio = packed record a_part: 0 .. 31;  { the quantity a + 16 in our derivation }
    b_part: 0 .. 31;  { the quantity b in our derivation }
    c_part: 0 .. '100000;  { the quantity c in our derivation }
    end;
  scaled = integer;  { this data type is used for quantities in spt units }
```

This code is used in section 2.

7. The real problem is to define the procedures that TEX needs to deal with such *glue_ratio* values: (a) Given scaled numbers s, t, and y as above, to compute the corresponding *glue_ratio*. (b) Given a scaled number x and a *glue_ratio* g, to compute the scaled number $f(x)$. (c) Given a *glue_ratio* g, to print out a decimal equivalent of g for diagnostic purposes.

8. **Glue multiplication.** The easiest procedure of the three just mentioned is the one that is needed most often, namely, the computation of $f(x)$.

PASCAL doesn't have built-in binary shift commands or built-in exponentiation, although many computers do have this capability. Therefore our arithmetic routines use an array called '*two_to_the*', containing powers of two. Divisions by powers of two are never done in the programs below when the dividend is negative, so the operations can safely be replaced by right shifts on machines for which this is most appropriate. (Contrary to popular opinion, the PASCAL operation 'x **div** 2' is not the same as shifting x right one binary place, when x is a negative odd integer, if the computer uses two's complement arithmetic. But division is equivalent to shifting when x is nonnegative.)

```
⟨Globals in the outer block 8⟩ ≡
two_to_the: array [0 .. 30] of integer;   { two_to_the[k] = $2^k$ }
```

See also sections 15 and 20.
This code is used in section 2.

```
9.  ⟨Local variables for initialization 9⟩ ≡
k: 1 .. 30;   { an index for initializing two_to_the }
```

This code is used in section 2.

```
10.  ⟨Set initial values 10⟩ ≡
  two_to_the[0] ← 1;
  for k ← 1 to 30 do two_to_the[k] ← two_to_the[k − 1] + two_to_the[k − 1];
```

This code is used in section 2.

11. The glue-multiplication function f can now be written:

```
  define ga ≡ g.a_part   { convenient abbreviations }
  define gb ≡ g.b_part   { as alternatives to }
  define gc ≡ g.c_part   { PASCAL's with statement }
function glue_mult(x : scaled; g : glue_ratio): integer;   { returns f(x) as above, assuming that x ≥ 0 }
  begin if ga > 16 then x ← x div two_to_the[ga − 16]   { right shift by a places }
  else x ← x * two_to_the[16 − ga];   { left shift by −a places }
  glue_mult ← (x * gc) div two_to_the[gb];   { right shift cx by b places }
  end;
```

12. Glue setting. The *glue_fix* procedure computes a, b, and c by the method explained above. TEX does not normally compute the quantity y, but it would not be difficult to make it do so.

This procedure would be a function that returns a *glue_ratio*, if PASCAL would allow functions to produce records as values.

```
procedure glue_fix(s, t, y : scaled; var g : glue_ratio);
  var a, b, c: integer;  {components of the desired ratio}
    k, h: integer;  {$30 - \lfloor \lg s \rfloor$, $30 - \lfloor \lg t \rfloor$}
    ss: integer;  {original (unnormalized) value of s}
    q, r, v: integer;  {quotient, remainder, divisor}
    w: integer;  {$2^l$}
  begin ⟨Normalize s, t, and y, computing a, k, and h 13⟩;
  if $t < s$ then $b \leftarrow 15 - a - k + h$ else $b \leftarrow 14 - a - k + h$;
  if $b < 0$ then
    begin write_ln('!␣Excessive␣glue.');  {error message}
    $b \leftarrow 0$; $c \leftarrow 1$;  {make $f(x) = \lfloor 2^{-a}\dot{x} \rfloor$}
    end
  else begin if $k \geq 16$ then  {easy case, $s < 2^{15}$}
      $c \leftarrow (t$ div two_to_the$[h - a - b] + ss - 1)$ div ss
    else ⟨Compute c by long division 14⟩;
    end;
  $ga \leftarrow a + 16$; $gb \leftarrow b$; $gc \leftarrow c$;
  end;
```

13. ⟨Normalize s, t, and y, computing a, k, and h 13⟩ ≡

```
  begin $a \leftarrow 15$; $k \leftarrow 0$; $h \leftarrow 0$; $ss \leftarrow s$;
  while $y <$ '10000000000 do  {y is known to be positive}
    begin decr(a); $y \leftarrow y + y$;
    end;
  while $s <$ '10000000000 do  {s is known to be positive}
    begin incr(k); $s \leftarrow s + s$;
    end;
  while $t <$ '10000000000 do  {t is known to be positive}
    begin incr(h); $t \leftarrow t + t$;
    end;
  end
```

This code is used in section 12.

14. ⟨Compute c by long division 14⟩ ≡

```
  begin $w \leftarrow$ two_to_the$[16 - k]$; $v \leftarrow ss$ div $w$; $q \leftarrow t$ div $v$; $r \leftarrow ((t \bmod v) * w) - ((ss \bmod w) * q)$;
  if $r > 0$ then
    begin incr(q); $r \leftarrow r - ss$;
    end
  else while $r \leq -ss$ do
      begin decr(q); $r \leftarrow r + ss$;
      end;
  if $a + b + k - h = -17$ then $c \leftarrow (q + 1)$ div 2  {$l = 16 + k - h$}
  else $c \leftarrow (q + 3)$ div 4;
  end
```

This code is used in section 12.

15. Glue-set printing. The last of the three procedures we need is *print_glue*, which displays a *glue_ratio* in symbolic decimal form. Before constructing such a procedure, we shall consider some simpler routines, copying them from TEX.

```
  define unity ≡ '200000   { $2^{16}$, represents 1.0000 }
⟨Globals in the outer block 8⟩ +≡
dig: array [0 .. 15] of 0 .. 9;   { for storing digits }
```

16. An array of digits is printed out by *print_digs*.

```
procedure print_digs(k : integer);   { prints dig[$k-1$] ... dig[0] }
  begin while $k > 0$ do
    begin decr(k); write(chr(ord('0') + dig[k]));
    end;
  end;
```

17. A nonnegative integer is printed out by *print_int*.

```
procedure print_int(n : integer);   { prints an integer in decimal form }
  var k: 0 .. 12;   { index to current digit; we assume that $0 \le n < 10^{12}$ }
  begin k ← 0;
  repeat dig[k] ← n mod 10; n ← n div 10; incr(k);
  until $n = 0$;
  print_digs(k);
  end;
```

18. And here is a procedure to print a nonnegative *scaled* number.

```
procedure print_scaled(s : scaled);   { prints a scaled real, truncated to four digits }
  var k: 0 .. 3;   { index to current digit of the fraction part }
  begin print_int(s div unity);   { print the integer part }
  s ← ((s mod unity) * 10000) div unity;
  for k ← 0 to 3 do
    begin dig[k] ← s mod 10; s ← s div 10;
    end;
  write('.'); print_digs(4);
  end;
```

19. Now we're ready to print a *glue_ratio*. Since the effective multiplier is $2^{-a-b}c$, we will display the scaled integer $2^{16-a-b}c$, taking care to print something special if this quantity is terribly large.

```
procedure print_glue(g : glue_ratio);   { prints a glue multiplier }
  var d: −32 .. 31;   { the quantity $16 - a - b$ }
  begin d ← 32 − ga − gb;   { the amount to shift c }
  while $d > 15$ do
    begin write('2x'); decr(d);   { indicate multiples of 2 for BIG cases }
    end;
  if $d < 0$ then print_scaled(gc div two_to_the[−d])   { shift right }
  else print_scaled(gc * two_to_the[d])   { shift left }
  end;
```

20. The driver program. In order to test these routines, we will assume that the *input* file contains a sequence of test cases, where each test case consists of the integer numbers t, $x_1, \ldots, x_n$, 0; the final test case should be followed by an additional zero.

```
⟨Globals in the outer block 8⟩ +≡
x: array [1 .. 1000] of scaled;   { the $x_i$ }
t: scaled;   { the desired total }
m: integer;   { the test case number }
```

21. Each case will be processed by the following routine, which assumes that t has already been read.

```
procedure test;   { processes the next data set, given $t$ and $m$ }
  var n: 0 .. 1000;   { the number of items }
    k: 0 .. 1000;   { runs through the items }
    y: scaled;   { $\max_{1\le i\le n}|x_1|$ }
    g: glue_ratio;   { the computed glue multiplier }
    s: scaled;   { the sum $x_1 + \cdots + x_n$ }
    ts: scaled;   { the sum $f(x_1) + \cdots + f(x_n)$ }
  begin write_ln('Test␣data␣set␣number␣', m : 0, ':');
  ⟨Read $x_1, \ldots, x_n$ 22⟩;
  ⟨Compute $s$ and $y$ 23⟩;
  if $s \le 0$ then write_ln('Invalid␣data␣(nonpositive␣sum);␣this␣set␣rejected.')
  else begin ⟨Compute $g$ and print it 24⟩;
    ⟨Print the values of $x_i$, $f(x_i)$, and the totals 25⟩;
    end;
  end;
```

```
22. ⟨Read $x_1, \ldots, x_n$ 22⟩ ≡
  begin n ← 0;
  repeat incr(n); read(x[n]);
  until x[n] = 0;
  decr(n);
  end
```

This code is used in section 21.

```
23. ⟨Compute $s$ and $y$ 23⟩ ≡
  begin s ← 0; y ← 0;
  for k ← 1 to n do
    begin s ← s + x[k];
    if y < abs(x[k]) then y ← abs(x[k]);
    end;
  end
```

This code is used in section 21.

```
24. ⟨Compute $g$ and print it 24⟩ ≡
  begin glue_fix(s, t, y, g);   { set $g$, perhaps print an error message }
  write('␣␣Glue␣ratio␣is␣'); print_glue(g); write_ln('␣(', ga − 16 : 0, '.', gb : 0, ',', gc : 0, ')');
  end
```

This code is used in section 21.

25. ⟨Print the values of x_i, $f(x_i)$, and the totals 25⟩ ≡

```
begin ts ← 0;
for k ← 1 to n do
  begin write(x[k] : 20);
  if x[k] ≥ 0 then y ← glue_mult(x[k], g) else y ← −glue_mult(−x[k], g);
  write_ln(y : 15); ts ← ts + y;
  end;
write_ln('␣Totals', s : 13, ts : 15, '␣(versus␣', t : 0, ')');
end
```

This code is used in section 21.

26. Here is the main program.

```
begin initialize; m ← 1; read(t);
while t > 0 do
  begin test; incr(m); read(t);
  end;
end.
```

27. Index. Here are the section numbers where various identifiers are used in the program, and where various topics are discussed.

a: 12.
a_part: 6, 11.
abs: 23.
b: 12.
b_part: 6, 11.
c: 12.
c_part: 6, 11.
chr: 16.
d: 19.
decr: 3, 13, 14, 16, 19, 22.
dig: 15, 16, 17, 18.
div: 8.
error analysis: 4.
g: 11, 12, 21.
ga: 11, 12, 19, 24.
gb: 11, 12, 19, 24.
gc: 11, 12, 19, 24.
GLUE: 2.
glue_fix: 12, 24.
glue_mult: 11, 25.
glue_ratio: 6, 7, 11, 12, 15, 19, 21.
h: 12.
hairy mathematics: 4.
incr: 3, 13, 14, 17, 22, 26.
initialize: 2, 26.
input: 2, 20.
integer: 6, 8, 11, 12, 16, 17, 20.
k: 12, 17, 18.
m: 20.
n: 21.
ord: 16.
output: 2.
print_digs: 16, 17, 18.
print_glue: 15, 19, 24.
print_int: 17, 18.
print_scaled: 18, 19.
program header: 2.
q: 12.
r: 12.
read: 22, 26.
real: 6.
s: 12, 21.
scaled: 6, 11, 12, 18, 20, 21.
shifting: 8.
ss: 12, 13, 14.
t: 12, 20.
test: 21, 26.
ts: 21, 25.
two_to_the: 8, 9, 10, 11, 12, 14, 19.
unity: 15, 18.
v: 12.
w: 12.
with: 11.
write: 16, 18, 19, 24, 25.
write_ln: 12, 21, 24, 25.
x: 11, 20.
y: 12, 21.

⟨Compute c by long division 14⟩

⟨Compute g and print it 24⟩

⟨Compute s and y 23⟩

⟨Globals in the outer block 8⟩

⟨Local variables for initialization 9⟩

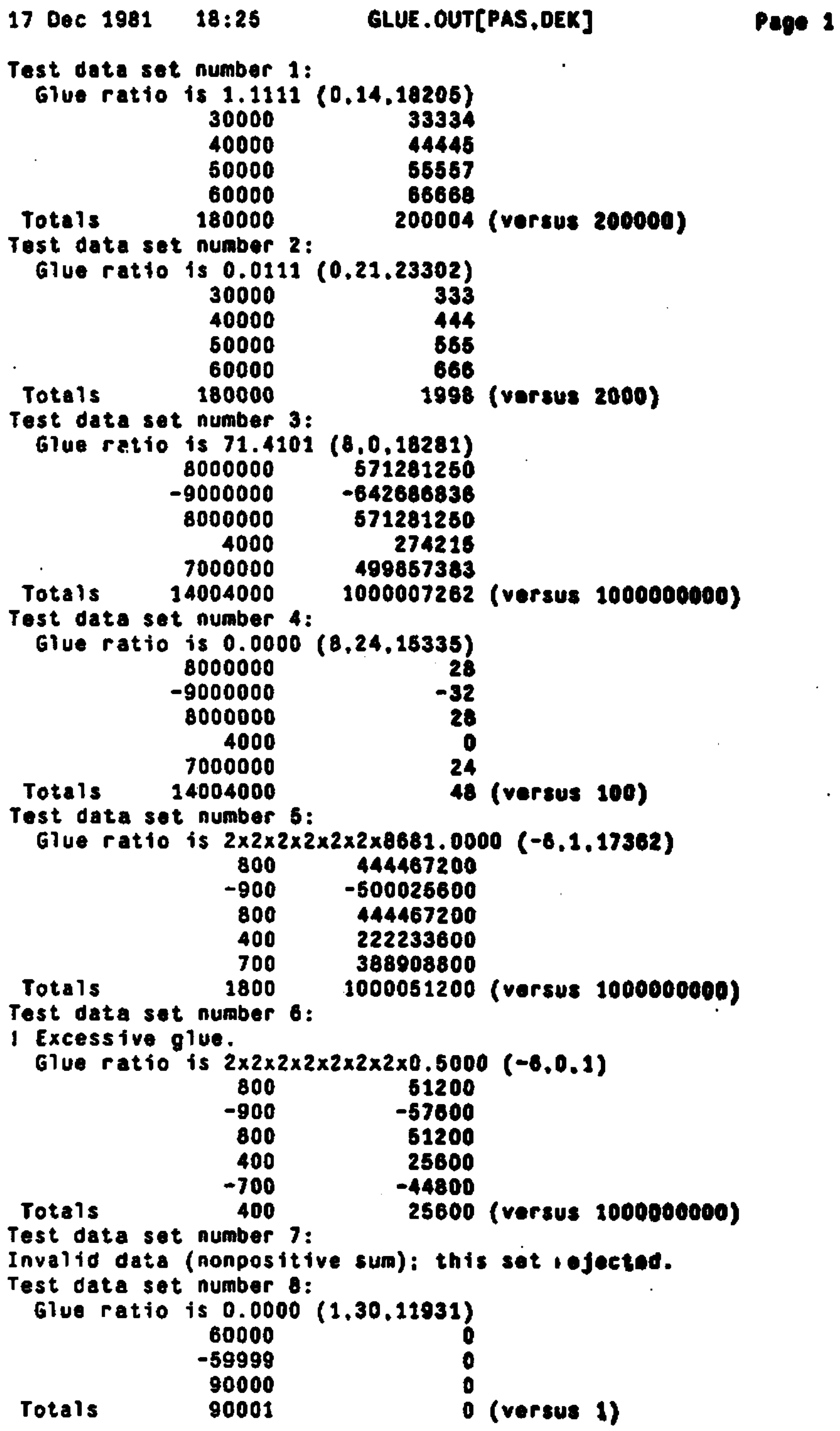
⟨Normalise s, t, and y, computing a, k, and h 13⟩

⟨Print the values of x_i, $f(x_i)$, and the totals 25⟩

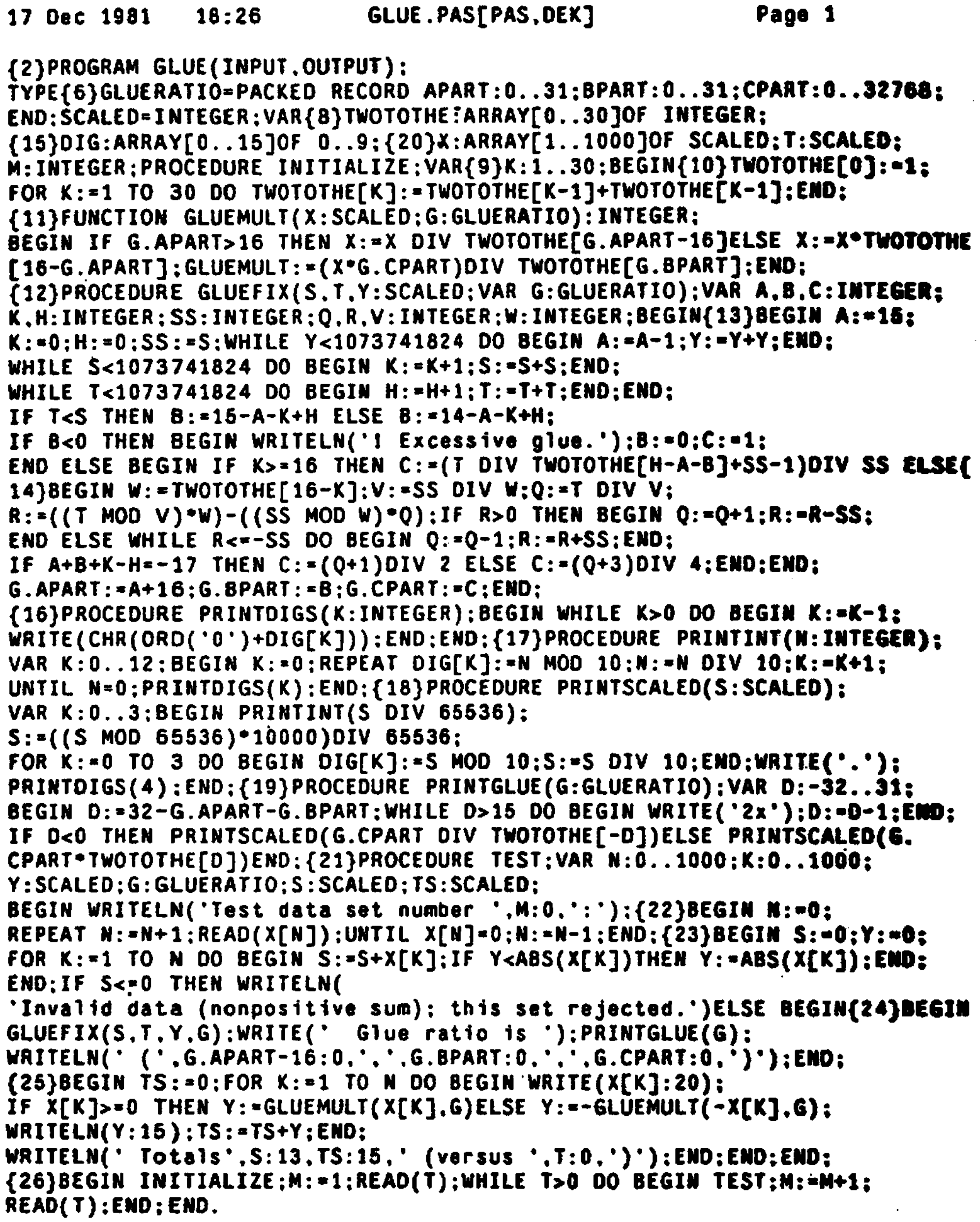
⟨Read $x_1, \ldots, x_n$ 22⟩

⟨Set initial values 10⟩

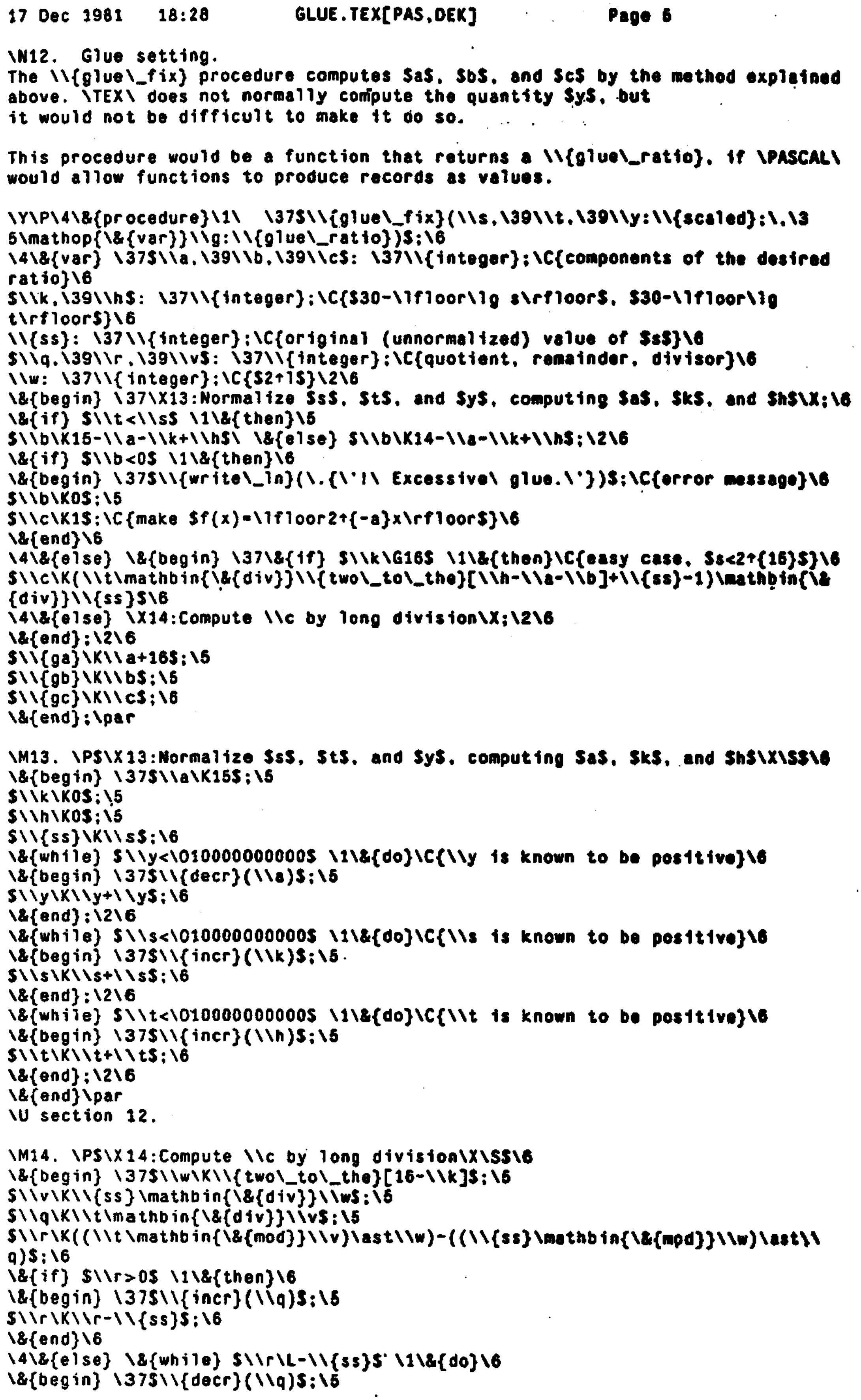
⟨Types in the outer block 6⟩



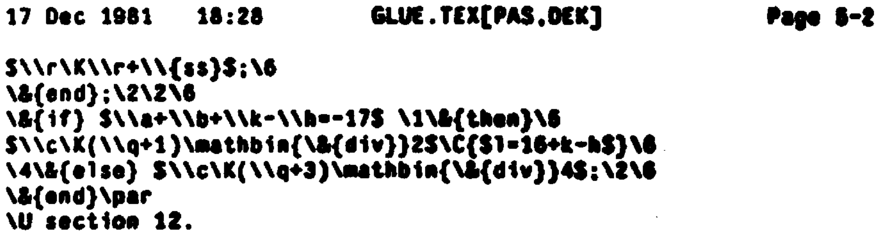
```
17 Dec 1981   18:25        GLUE.OUT[PAS,DEK]          Page 1

Test data set number 1:
  Glue ratio is 1.1111 (0,14,18205)
               30000          33334
               40000          44445
               50000          55557
               60000          66668
 Totals       180000         200004 (versus 200000)
Test data set number 2:
  Glue ratio is 0.0111 (0,21,23302)
               30000            333
               40000            444
               50000            555
               60000            666
 Totals       180000           1998 (versus 2000)
Test data set number 3:
  Glue ratio is 71.4101 (8,0,18281)
             8000000      571281250
            -9000000     -642686836
             8000000      571281250
                4000         274215
             7000000      499857383
 Totals     14004000     1000007262 (versus 1000000000)
Test data set number 4:
  Glue ratio is 0.0000 (8,24,15335)
             8000000             28
            -9000000            -32
             8000000             28
                4000              0
             7000000             24
 Totals     14004000             48 (versus 100)
Test data set number 5:
  Glue ratio is 2x2x2x2x2x2x8681.0000 (-6,1,17362)
                 800      444467200
                -900     -500025600
                 800      444467200
                 400      222233600
                 700      388908800
 Totals         1800     1000051200 (versus 1000000000)
Test data set number 6:
! Excessive glue.
  Glue ratio is 2x2x2x2x2x2x2x0.5000 (-6,0,1)
                 800          51200
                -900         -57600
                 800          51200
                 400          25600
                -700         -44800
 Totals          400          25600 (versus 1000000000)
Test data set number 7:
Invalid data (nonpositive sum); this set rejected.
Test data set number 8:
  Glue ratio is 0.0000 (1,30,11931)
               60000              0
              -59999              0
               90000              0
 Totals        90001              0 (versus 1)
```



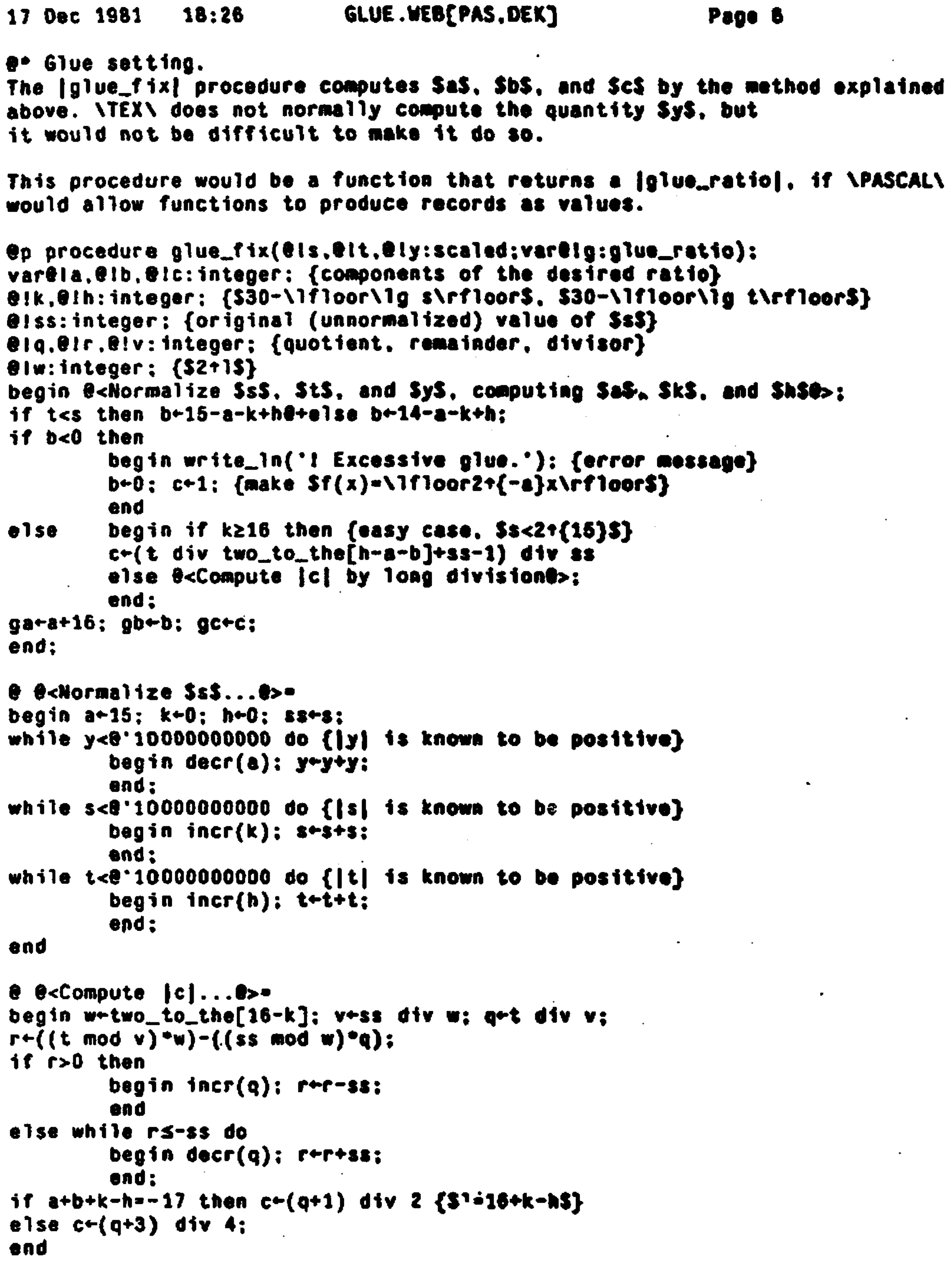
```
17 Dec 1981   18:26         GLUE.PAS[PAS,DEK]             Page 1

{2}PROGRAM GLUE(INPUT,OUTPUT);
TYPE{6}GLUERATIO=PACKED RECORD APART:0..31;BPART:0..31;CPART:0..32768;
END;SCALED=INTEGER;VAR{8}TWOTOTHE:ARRAY[0..30]OF INTEGER;
{15}DIG:ARRAY[0..15]OF 0..9;{20}X:ARRAY[1..1000]OF SCALED;T:SCALED;
M:INTEGER;PROCEDURE INITIALIZE;VAR{9}K:1..30;BEGIN{10}TWOTOTHE[0]:=1;
FOR K:=1 TO 30 DO TWOTOTHE[K]:=TWOTOTHE[K-1]+TWOTOTHE[K-1];END;
{11}FUNCTION GLUEMULT(X:SCALED;G:GLUERATIO):INTEGER;
BEGIN IF G.APART>16 THEN X:=X DIV TWOTOTHE[G.APART-16]ELSE X:=X*TWOTOTHE
[16-G.APART];GLUEMULT:=(X*G.CPART)DIV TWOTOTHE[G.BPART];END;
{12}PROCEDURE GLUEFIX(S,T,Y:SCALED;VAR G:GLUERATIO);VAR A,B,C:INTEGER;
K,H:INTEGER;SS:INTEGER;Q,R,V:INTEGER;W:INTEGER;BEGIN{13}BEGIN A:=15;
K:=0;H:=0;SS:=S;WHILE Y<1073741824 DO BEGIN A:=A-1;Y:=Y+Y;END;
WHILE S<1073741824 DO BEGIN K:=K+1;S:=S+S;END;
WHILE T<1073741824 DO BEGIN H:=H+1;T:=T+T;END;END;
IF T<S THEN B:=15-A-K+H ELSE B:=14-A-K+H;
IF B<0 THEN BEGIN WRITELN('! Excessive glue.');B:=0;C:=1;
END ELSE BEGIN IF K>=16 THEN C:=(T DIV TWOTOTHE[H-A-B]+SS-1)DIV SS ELSE{
14}BEGIN W:=TWOTOTHE[16-K];V:=SS DIV W;Q:=T DIV V;
R:=((T MOD V)*W)-((SS MOD W)*Q);IF R>0 THEN BEGIN Q:=Q+1;R:=R-SS;
END ELSE WHILE R<=-SS DO BEGIN Q:=Q-1;R:=R+SS;END;
IF A+B+K-H=-17 THEN C:=(Q+1)DIV 2 ELSE C:=(Q+3)DIV 4;END;END;
G.APART:=A+16;G.BPART:=B;G.CPART:=C;END;
{16}PROCEDURE PRINTDIGS(K:INTEGER);BEGIN WHILE K>0 DO BEGIN K:=K-1;
WRITE(CHR(ORD('0')+DIG[K]));END;END;{17}PROCEDURE PRINTINT(N:INTEGER);
VAR K:0..12;BEGIN K:=0;REPEAT DIG[K]:=N MOD 10;N:=N DIV 10;K:=K+1;
UNTIL N=0;PRINTDIGS(K);END;{18}PROCEDURE PRINTSCALED(S:SCALED);
VAR K:0..3;BEGIN PRINTINT(S DIV 65536);
S:=((S MOD 65536)*10000)DIV 65536;
FOR K:=0 TO 3 DO BEGIN DIG[K]:=S MOD 10;S:=S DIV 10;END;WRITE('.');
PRINTDIGS(4);END;{19}PROCEDURE PRINTGLUE(G:GLUERATIO);VAR D:-32..31;
BEGIN D:=32-G.APART-G.BPART;WHILE D>15 DO BEGIN WRITE('2x');D:=D-1;END;
IF D<0 THEN PRINTSCALED(G.CPART DIV TWOTOTHE[-D])ELSE PRINTSCALED(G.
CPART*TWOTOTHE[D])END;{21}PROCEDURE TEST;VAR N:0..1000;K:0..1000;
Y:SCALED;G:GLUERATIO;S:SCALED;TS:SCALED;
BEGIN WRITELN('Test data set number ',M:0,':');{22}BEGIN N:=0;
REPEAT N:=N+1;READ(X[N]);UNTIL X[N]=0;N:=N-1;END;{23}BEGIN S:=0;Y:=0;
FOR K:=1 TO N DO BEGIN S:=S+X[K];IF Y<ABS(X[K])THEN Y:=ABS(X[K]);END;
END;IF S<=0 THEN WRITELN(
'Invalid data (nonpositive sum); this set rejected.')ELSE BEGIN{24}BEGIN
GLUEFIX(S,T,Y,G);WRITE('  Glue ratio is ');PRINTGLUE(G);
WRITELN(' (',G.APART-16:0,',',G.BPART:0,',',G.CPART:0,')');END;
{25}BEGIN TS:=0;FOR K:=1 TO N DO BEGIN WRITE(X[K]:20);
IF X[K]>=0 THEN Y:=GLUEMULT(X[K],G)ELSE Y:=-GLUEMULT(-X[K],G);
WRITELN(Y:15);TS:=TS+Y;END;
WRITELN(' Totals',S:13,TS:15,' (versus ',T:0,')');END;END;END;
{26}BEGIN INITIALIZE;M:=1;READ(T);WHILE T>0 DO BEGIN TEST;M:=M+1;
READ(T);END;END.
```



```
17 Dec 1981   18:28        GLUE.TEX[PAS,DEK]           Page 5

\N12.  Glue setting.
The \\{glue\_fix} procedure computes $a$, $b$, and $c$ by the method explained
above. \TEX\ does not normally compute the quantity $y$, but
it would not be difficult to make it do so.

This procedure would be a function that returns a \\{glue\_ratio}, if \PASCAL\
would allow functions to produce records as values.

\Y\P\4\&{procedure}\1\  \37$\\{glue\_fix}(\\s,\39\\t,\39\\y:\\{scaled};\,\3
5\mathop{\&{var}}\\g:\\{glue\_ratio})$;\6
\4\&{var} \37$\\a,\39\\b,\39\\c$: \37\\{integer};\C{components of the desired
ratio}\6
$\\k,\39\\h$: \37\\{integer};\C{$30-\lfloor\lg s\rfloor$, $30-\lfloor\lg
t\rfloor$}\6
\\{ss}: \37\\{integer};\C{original (unnormalized) value of $s$}\6
$\\q,\39\\r,\39\\v$: \37\\{integer};\C{quotient, remainder, divisor}\6
\\w: \37\\{integer};\C{$2↑1$}\2\6
\&{begin} \37\X13:Normalize $s$, $t$, and $y$, computing $a$, $k$, and $h$\X;\6
\&{if} $\\t<\\s$ \1\&{then}\5
$\\b\K15-\\a-\\k+\\h$\ \&{else} $\\b\K14-\\a-\\k+\\h$;\2\6
\&{if} $\\b<0$ \1\&{then}\6
\&{begin} \37$\\{write\_ln}(\.{\'!\ Excessive\ glue.\'})$;\C{error message}\6
$\\b\K0$;\5
$\\c\K1$;\C{make $f(x)=\lfloor2↑{-a}x\rfloor$}\6
\&{end}\6
\4\&{else} \&{begin} \37\&{if} $\\k\G16$ \1\&{then}\C{easy case, $s<2↑{15}$}\6
$\\c\K(\\t\mathbin{\&{div}}\\{two\_to\_the}[\\h-\\a-\\b]+\\{ss}-1)\mathbin{\&
{div}}\\{ss}$\6
\4\&{else} \X14:Compute \\c by long division\X;\2\6
\&{end};\2\6
$\\{ga}\K\\a+16$;\5
$\\{gb}\K\\b$;\5
$\\{gc}\K\\c$;\6
\&{end};\par

\M13. \P$\X13:Normalize $s$, $t$, and $y$, computing $a$, $k$, and $h$\X\S$\6
\&{begin} \37$\\a\K15$;\5
$\\k\K0$;\5
$\\h\K0$;\5
$\\{ss}\K\\s$;\6
\&{while} $\\y<\O10000000000$ \1\&{do}\C{\\y is known to be positive}\6
\&{begin} \37$\\{decr}(\\a)$;\5
$\\y\K\\y+\\y$;\6
\&{end};\2\6
\&{while} $\\s<\O10000000000$ \1\&{do}\C{\\s is known to be positive}\6
\&{begin} \37$\\{incr}(\\k)$;\5
$\\s\K\\s+\\s$;\6
\&{end};\2\6
\&{while} $\\t<\O10000000000$ \1\&{do}\C{\\t is known to be positive}\6
\&{begin} \37$\\{incr}(\\h)$;\5
$\\t\K\\t+\\t$;\6
\&{end};\2\6
\&{end}\par
\U section 12.

\M14. \P$\X14:Compute \\c by long division\X\S$\6
\&{begin} \37$\\w\K\\{two\_to\_the}[16-\\k]$;\6
$\\v\K\\{ss}\mathbin{\&{div}}\\w$;\5
$\\q\K\\t\mathbin{\&{div}}\\v$;\5
$\\r\K((\\t\mathbin{\&{mod}}\\v)\ast\\w)-((\\{ss}\mathbin{\&{mod}}\\w)\ast\\
q)$;\6
\&{if} $\\r>0$ \1\&{then}\6
\&{begin} \37$\\{incr}(\\q)$;\5
$\\r\K\\r-\\{ss}$;\6
\&{end}\6
\4\&{else} \&{while} $\\r\L-\\{ss}$ \1\&{do}\6
\&{begin} \37$\\{decr}(\\q)$;\5
```



```
17 Dec 1981   18:28          GLUE.TEX[PAS,DEK]            Page 8-2

$\\r\K\\r+\\{ss}$;\6
\&{end};\2\2\6
\&{if} $\\a+\\b+\\k-\\h=-17$ \1\&{then}\6
$\\c\K(\\q+1)\mathbin{\&{div}}2$\C{$l=16+k-h$}\6
\4\&{else} $\\c\K(\\q+3)\mathbin{\&{div}}4$;\2\6
\&{end}\par
\U section 12.
```



```
17 Dec 1981   18:26         GLUE.WEB[PAS,DEK]            Page 6

@* Glue setting.
The |glue_fix| procedure computes $a$, $b$, and $c$ by the method explained
above. \TEX\ does not normally compute the quantity $y$, but
it would not be difficult to make it do so.

This procedure would be a function that returns a |glue_ratio|, if \PASCAL\
would allow functions to produce records as values.

@p procedure glue_fix(@!s,@!t,@!y:scaled;var@!g:glue_ratio);
var@!a,@!b,@!c:integer; {components of the desired ratio}
@!k,@!h:integer; {$30-\lfloor\lg s\rfloor$, $30-\lfloor\lg t\rfloor$}
@!ss:integer; {original (unnormalized) value of $s$}
@!q,@!r,@!v:integer; {quotient, remainder, divisor}
@!w:integer; {$2↑1$}
begin @<Normalize $s$, $t$, and $y$, computing $a$, $k$, and $h$@>;
if t<s then b←15-a-k+h@+else b←14-a-k+h;
if b<0 then
        begin write_ln('! Excessive glue.'); {error message}
        b←0; c←1; {make $f(x)=\lfloor2↑{-a}x\rfloor$}
        end
else    begin if k≥16 then {easy case, $s<2↑{15}$}
        c←(t div two_to_the[h-a-b]+ss-1) div ss
        else @<Compute |c| by long division@>;
        end;
ga←a+16; gb←b; gc←c;
end;

@ @<Normalize $s$...@>=
begin a←15; k←0; h←0; ss←s;
while y<@'10000000000 do {|y| is known to be positive}
        begin decr(a); y←y+y;
        end;
while s<@'10000000000 do {|s| is known to be positive}
        begin incr(k); s←s+s;
        end;
while t<@'10000000000 do {|t| is known to be positive}
        begin incr(h); t←t+t;
        end;
end

@ @<Compute |c|...@>=
begin w←two_to_the[16-k]; v←ss div w; q←t div v;
r←((t mod v)*w)-((ss mod w)*q);
if r>0 then
        begin incr(q); r←r-ss;
        end
else while r≤-ss do
        begin decr(q); r←r+ss;
        end;
if a+b+k-h=-17 then c←(q+1) div 2 {$'=16+k-h$}
else c←(q+3) div 4;
end
```

* * * * * * * * * * *

Site Reports

* * * * * * * * * * *

CDC TEX

Michael J. Frisch
University of Minnesota

I have decided to give up working on the Minnesota version of TEX because the version running at the Regional EDP Center at the University of Aarhus (RECAU) in Denmark is much smaller and easier to change. Erik Bertelsen of RECAU has agreed to become the site coordinator for TEX on CDC machines. His version of TEX works quite well and is fully debugged. It runs in about 38K of memory and uses about 2 CP seconds per page on the RECAU Cyber 173. It is based on the May, 1980 version of Stanford TEX but will be updated when TEX 82 becomes available.

However, Erik's version of TEX uses the RECAU Pascal library feature which is not a part of Pascal 6000 systems for CDC computers. Erik plans to revise his source code so that his version of TEX can be compiled at other CDC sites. He has installed a binary version at Minnesota using temporary revisions.

Presently, device drivers are available at RECAU for daisy wheel printers of the Qume/Diablo/NEC type, for the Compugraphics Unisetter typesetter, for Tektronix 4014 terminals emulating the Unisetter using the DISSPLA graphics package from ISSCO, and for any other devices that use DISSPLA's software defined characters. At Minnesota, I have been working on a Varian 200 dot per inch plotter driver and I plan to revise it to work with Erik's TEX.

There are some minimal requirements for being able to install the source of Erik's revised TEX. The Pascal compiler is needed, of course. The CDC UPDATE utility must be at least at correction level 528 because Erik's TEX uses the long input lines and 8-in-12 ASCII character set features of UPDATE. Note that a more recent version of UPDATE will probably be required since bugs in level 528 can sometimes prevent installation of TEX. Because of the different character sets, CDC users must also have a way to read and print 8-in-12 files if they work with the source code of Erik's TEX. This is easily done in the NOS operating system at correction level 509 and above, or else the Pascal COPYCH utility can be used.

Erik has written several programs that simplify the creation of font information files for the various devices that can be used with TEX. He has added the Stanford TFX files for 200 per dot inch plotters to his library of fonts so that the Minnesota plotter driver can be used when completed. Users of devices other than the ones mentioned above must create their own font information files and must write device drivers. Erik's code provides good examples for commonly-used devices.

CDC sites interested in the RECAU version of TEX should contact Erik for details about distribution. His address and phone/Telex are:

Erik Bertelsen
The Regional EDP Center
University of Aarhus (RECAU)
Ny Munkegade
Bygning 540
DK-8000 Aarhus C
Denmark
International Phone: 45 6 128355
Telex: 64 754 recau dk

* * * * * * * * * * *

HP3000 SITE REPORT

Lance Carnes

Just this month the number of HP3000 TEX users has doubled—a copy has been installed at another HP3000 site. After the upcoming HP3000 International Users Meeting in San Antonio, where I will be conducting demonstrations and in general promoting TEX, there should be more users.

This implementation is still rather slow. It consumes 5 to 10 cpu seconds, and 20 to 30 elapsed seconds, per page of output. However, a TEX run may be submitted to a batch job, which will execute in background while foreground users are editing, compiling or eating lunch. In this way the user may avoid waiting for long periods of time in front of his or her terminal, only to find that "`! You can't do that in restricted horizontal mode`", or whatever.

(Slow is relative, however. Last week while working at Stanford I was able to compile 6 pages in 4 hours. The SAIL machine was in poor health and would die midway through an edit or TEX compile, and stay down for 20 to 30 minutes. Give me a slow machine that keeps running any day!)

There is hope for speeding up this implementation, though. All I need is some time to study where the bottlenecks are and optimize around them. Unfortunately, I do not currently earn my living by computerized typesetting, nor do I have a sponsor for my TEXploits. Time must be stolen from my evenings and weekends to apply the necessary touches to this implementation.

The Hewlett-Packard Boise Division is hard at work on a device driver for the HP2680 Laser Printer. At a resolution of 180 dots/inch this will provide a useful device for printing training manuals, technical documentation and other similar materials.

* * * * * * * * * * *

TEX AT UNIVERSITY OF MILAN

G. Canzii
D. Lucarella
A. Pilenga

Since November 1981, PASCAL TEX runs succesfully on IBM computer, series 3032, under MVS operating system in a batch environment. Output is on VERSATEC printer-plotter off-line. TEX has been compiled by "PASCAL 8000" release 2. The TEX release was received from Stanford University through I. Zabala on October 1980, using TFX.

The following transport problems have been fixed:

- "PASCAL 8000" does not accept more than one second level EXTERNAL procedure, therefore all SYSDEP routines were loaded respectively in "TEXPRE" and "TEX".
- To pass the compilation under "PASCAL 8000" intervention on syntax of some statement was required.
- TEX works with ASCII characters while IBM 3032 with EBCDIC code and therefore we inserted two routines "`EBCDICTOASCII`" and "`ASCIITOEBCDIC`" to obtain I/O activity on our system.
- Run-time dynamic allocation of files is not allowed under MVS operating system. Consequently, and to optimize portability, we decided for static allocation of data sets.
  To do that, it was necessary to implement two procedures:
  a) an initialization procedure that loads all filenames;
  b) a procedure that, reading the font-names supplied by the user, opens the corresponding file.
- "`DRIVER`" program for VERSATEC device has been rewritten.

The following problems have been met in the "preprocessor" and "processor" modules.

Since SYSDEP was split, our discussion includes it. "TEXPRE" required these modifications:

1) "PASCAL 8000" does not initialize variables, consequently the "`Procedure INITPROCEDURE`" is called, as first, at the beginning of module.
2) We have written the "`Procedure REFRESH`" that initializes those variables not present in "`INITPROCEDURE`".
3) All files have been declared as text-files.
4) In the "`Procedure GETNEXT`" we have added the global variable "`FONTE`", to store the fontname supplied by the user.

"TEX" required the following modifications:

1) The same intervention already listed at points (1), (2) and (3) for "TEXPRE".
2) "PASCAL 8000" generates real numbers only in double precision (2 words). It was necessary to reformat "TBLFIL" file that now contains variable length records. Consequently, routines to access this file in read/write have been changed.
3) Differing from Stanford, in our version the basic fonts have been pre-defined in the preprocessor module. Therefore, the "`Procedure DEFINEFONT`" stores font information exclusively in the static portion of TEX memory. We have written the "`Procedure SPACE`" to load into dynamic portion of memory the space values for every font.
4) In the "`HYPHENATIONWORD`", besides existing initializations, we provided
   "`TRUNCWORD (.0.):=0`" in "`LEXICALORDER`".

As future plans, we want to implement following modules:

a) a procedure to obtain automatic syllabic division for Italian language;
b) output driver for interfacing graphic terminal, model "TEKTRONIX 4014".

As soon as possible this version of TEX on IBM machines should be available for distribution followed by a detailed documentation. For information, contact:

Giovanni Canzii
Istituto di Cibernetica
Via Viotti 3/5
20133 Milano, Italy

Thanks are due to the Stanford group for suggestions, to Prof. Giovanni Degli Antoni for his support, and to Communication and Programming Project (between Honeywell Information Systems Italia and University of Milan-Institute of Cybernetics) for sponsoring our activity.

* * * * * * * * * * *

TeX-news from Pisa

by

L. Aiello
IEI, CNR

S. Pavan
ISI, University

This is an example of our TeX-output

TeX runs in Pisa in PASCAL-VS on the IBM 370/3033 of CNUCE (an Institute of the National Research Council) under VM/CMS. At CNUCE we have had access to a graphic station consisting of two screens: an alphanumeric IBM 3277 and a high resolution display TEKTRONIX 619, interconnected via the Display Graphic Attachement RPQ 7H0284. We have also had access to a VERSATEC, a (not very) high-resolution (160 pixels per inch) electrostatic printer. Drivers have been implemented both to output TeX-DVI files on the TEKTRONIX and on the VERSATEC. Hence, the markup file is displayed and edited on the IBM 3277, while the formatted pages (or parts thereof) are shown on the TEKTRONIX. Hardcopies of the final documents are produced on the VERSATEC.

TeX seems fully tuned up: in the last couple of months, no bug has been reported by the (still small) TeX-user community. We have already produced some documents (letters, a thesis, few technical papers) and they "look good".

We have developed some software around TeX, part in the form of TeX-macros, some in the form of PASCAL extensions. A source of inspiration in the development of our software has been SCRIBE. In particular, we have taken from SCRIBE the idea of "type of document".

We have designed and implemented a Preprocessor for TeXthat knows about a library of types of documents and can access a data base of bibliographic entries. The user, by simply invoking one of the predefined types, is guided in the composition of his document and is allowed to ignore all of the low level formatting commands. Among other things, the numbering of chapters, sections, subsections is performed automatically, as well as the construction of the table of contents, index and bibliography, provided the type of document contains such components.

In order to allow the Preprocessor to build the bibliography automatically we have designed and implemented an Editor. It is a simple data base

management system. It is guided by the structure of a type of bibliographic entry and facilitates the construction of the bibliographic data base in the form accepted by the Preprocessor.

Our software is somehow documented but, for the moment, in Italian.

In developing our software we have always had portability in mind, from two view points:

1) The Preprocessor and the Editor are written in PASCAL-VS; they are portable to other installations of TEX in PASCAL-VS.

2) The output of the Preprocessor is a legal TEX-input file; our TEX-able documents may be compiled by other TEX installations.

The reactions of the first users in Pisa is satisfactory. We are still very unhappy about the hardware we have: we hope to get soon a better printer.

Various people, besides us, have contributed at various stages of the TEX project in Pisa: G. Attardi, M. Cupidi, D. Dariol, P. Falsini, G. Prini. Dariol and Falsini made the installation happen, Aiello and Pavan took care of the Preprocessor and Editor.

VAX/VMS SITE REPORT

Monte C. Nichols

The users of TEX on VAX/VMS will be pleased to hear of the progress being made on their behalf, mostly through the courtesy and hard work of John Blair - Calma, Christopher Day - Lawrence Berkeley Laboratory, and David Kellerman - Oregon Software. Needless to say, none of this would have been possible without the initial work done by Barry Smith - Oregon Software, who has gone off on a well deserved sabbatical. To all these hard working guys—**THANKS!!!**

So what has happened you might ask? Well, while we are all anxiously waiting for TEX82 to become available, the aforementioned hard workers have been busy improving TEX as it exists on VAX/VMS. Most of the bugs have been fixed in the old version—which up to now has been TEX-Pascal as it existed in November 1980. In addition, they have added many features which make TEX a more usable VAX package. Such things as batch mode operation, logical directory names, ability to run TEX from any disk, and to have more than one TEX user on the system at any given time are some of the things that have been added. The Versatec spooler has also been modified several times and can now even be called upon to put out a discrete range of pages as output rather than always printing the entire document.

As if this weren't enough, David Kellerman is presently working on bringing up the most recent Stanford version of TEX-Pascal. This will of course be the version having magnification capability, etc. The new version should also get rid of the last few remaining bugs and keep the VAX/VMS community running in a superior fashion until TEX82 becomes available. David indicates this will be available by the time this issue of TUGboat reaches you. For those of you new to TUG, Oregon Software has volunteered to distribute TEX for the VAX/VMS community. For $50 they will send you a tape with all the VAX/TEX related files on it. See TUGboat Vol. 2 No. 2 for further information.

Both Chris and John have gotten WEB, TANGLE and WEAVE running on the VAX/VMS system, so we are in good shape to begin TEX82 as soon as it becomes available.

Finally, I have not sent out any special mailing to the VAX/VMS community as I discussed in an earlier site report. A number of folks have pointed out that news worth sending to those of us in the VMS community is probably of interest to other TEX users as well, and is best published in TUGboat. I would encourage anyone who has items of interest to submit them to TUGboat or to get in touch with me and I will mention them in the next VMS site report.

* * * * * * * * * * *

"small" TEX

* * * * * * * * * * *

Send submissions to:
Lance Carnes
163 Linden Lane
Mill Valley, CA 94941

This new column, dubbed small TEX by Barbara Beeton, will cover any implementations of TEX on small machines, whether actually in place or planned for the future. If you have already or are thinking of bringing up TEX on any machine smaller than a VAX or a DEC10, write and let us know what you are up to.

What is meant my a small machine? An HP3000 is not usually considered a small machine, although when I tried to bring up TEX-in-Pascal on it, it suddenly became small. A rule of thumb might be: if the address space of the machine is less than 20 bits, and it takes more than a Pascal compile to get TEX running, and your operating system does not support virtual memory, you have a small machine.

The current offerings in the small machine arena are:

- Onyx (Z8002-based system). See TUGboat Vol. 2 No. 2 "TEX on Small Machines", by Harris and McClure.
  This is not a TEX-in-Pascal implementation, but a rewrite in C under UNIX.
- HP3000 (16-bit minicomputer). See TUGboat Vol. 2 No. 3 "Hewlett-Packard HP3000 Site Report", by Lance Carnes.
  This is a TEX-in-Pascal implementation, using software-implemented virtual memory.

If any small machine implementations have been omitted, please accept my apologies, and let us know the details.

Rumor. (You shouldn't be reading this—you know how unreliable rumors are!) A well-known member of the original TEX team down on the Farm has indicated interest in bringing up a version on the M68000. Exactly when this will occur was not revealed by my source.

* * * * * * * * * * *

Warnings & Limitations

* * * * * * * * * * *

Another Hangup

In the last issue, you were warned that repetition of a \let statement can cause TeX to hang. It has been pointed out that \ifx can be used to detect recursion (provided you are using a recent enough version of TeX; see the errata list, extensions since June 30, 1981).

Another way to make TeX hang is to set to a negative value one of TeX's integer parameters that is expecting a positive value. For example, \chpar3 and \chpar13 (which will become \penpen, \dhpen and \adjpen in TeX82) have caused the SAIL version of TeX to loop at the Math Society (we are still running with a version of March 1981). This will probably be trapped in TeX82, but I didn't see it described in the differences list.

Barbara Beetor

* * * * * * * * * * *

MACRO COLUMN

Send Submissions to:
Lynne A. Price
TUG Macro Coordinator
Calma R&D
212 Gibraltar Dr.
Sunnyvale, CA 94086

In the last issue of TUGboat (Volume 2, No. 3) Michael Plass described macros for producing syntax diagrams. The package includes macros for automatically allocating box and counter numbers, much as Patrick Milligan's \DefineFont macro (TUGboat, Volume 2, No. 2) assigns font codes. Mike's allocation macros are

```
\def\Alloc#1#2{\def#2{#1}}
\def\AllocBox
 {\def\AllocBox{\def\AllocBox{\def\AllocBox
 {\def\AllocBox{\def\AllocBox{\def\AllocBox
 {\def\AllocBox{\def\AllocBox{\def\AllocBox
  {\Overflow
   }\Alloc9}\Alloc8}\Alloc7}\Alloc6}\Alloc5
    }\Alloc4}\Alloc3}\Alloc2}\Alloc1}
\def\AllocCtr
 {\def\AllocCtr{\def\AllocCtr
 {\def\AllocCtr{\def\AllocCtr
  {\Overflow}\Alloc8}\Alloc7}\Alloc6}\Alloc5}
```

An example of using these macros to select a counter number is

```
\AllocCtr\counternumber
\setcount\counternumber 0
```

The first time \AllocCtr is called, it executes the second \def\AllocCtr (thus redefining itself to be the text beginning with the third \def\AllocCtr and ending with Alloc 6) and calls Alloc with a first parameter of 5. The second time it is called, it redefines itself removing another nested definition and passing the value 6 to Alloc. Similarly, \AllocBox successively returns the digits 1 through 9.

* * * * * * * * * * *

TUGBOAT MACRO INDEX

The following list catalogues macros that have appeared in TUGboat. Entries are listed by volume, number, and page as well as author's name. Items that could not be categorized by an obvious headword have been listed under "miscellaneous". Many items refer to parts of large macro packages; users of other packages may find them valuable models for macros of their own.

Readers' comments on the format as well as the contents of this index are welcome.

ACM style	II:1	61, 82–83	A. Keller
Addresses	II:1	54	B. Beeton
. . . .	II:2	A-35	M. Díaz
Appendices	II:2	A-21	M. Díaz
Baseline, set to top of box .	II:1	60, 77	A. Keller
Bibliography	II:2	A-25	M. Díaz
Boxes	II:1	59, 73	A. Keller
Box numbers, automatic allocation	III:1	33	M. Plass
Branching, see If			
Capital letters			
large ~ at beginning of paragraph			
. . . . .	II:1	60, 78	A. Keller
. . . .	II:3	62	TeXarcana Class
. . . .	II:2	A-16	M. Díaz
Roman numerals . . .	II:1	120–121	P. Milligan, L. Price
Centering a sequence of lines .	II:2	A-13	M. Díaz

Chapters and Sections	II:1 60–61, 79–81	A. Keller
	II:1 111-118	L. Price
	II:2 A-8–9, 20-22	M. Díaz
Characters, macros to produce special	II:1 57, 67–70	A. Keller
Chemical notation	II:3 57–58	M. Nichols, B. Beeton
Columns		
balanced	II:3 58–59	L. Price
multiple	II:2 A-38–40	M. Díaz
	II:3 24–25	B. Beeton
Comparison of integral values	II:1 119–120	P. Milligan, L. Price
Counters		
automatic allocation	III:1 33	M. Plass
pseudo	II:1 60, 77	A. Keller
	II:1 120	P. Milligan, L. Price
Cross references	II:3 24	B. Beeton
Deferred output	II:1 60, 86–88	A. Keller
Division	II:2 47	B. McKay
Equality of integral values	II:1 119–120	P. Milligan, L. Price
Figures	II:2 A-25–27	M. Díaz
Font		
declaring families of a particular point size	II:1 56–57, 65–66	A. Keller
	II:2 A-11	M. Díaz
definition	II:1 119	P. Milligan, L. Price
	II:2 44–45	P. Milligan
display in table form	III:1 35	R. Beeman
Footnotes	II:1 58, 71–72	A. Keller
	II:2 A-24-25	M. Díaz
French	II:2 A-12	M. Díaz
Graphics	II:2 48–49	B. McKay
	II:3 63	TEXarcana Class
Headings, page	II:2 A-23–24	M. Díaz
Hidden Text	II:3 61	TEXarcana Class
If		
comparison of integral values	II:1 119–120	P. Milligan, L. Price
groupless \if	II:2 46	B. McKay
null string, see Null string		
testing math-style (display, script or scriptscript)	II:2 46	B. McKay
Index production	I:1 Appendix A	T. Winograd, W. Paxton
	II:2 A-28	M. Díaz
Justification		
of reviewer's names	II:3 62	TEXarcana Class
right ~	II:3 63	TEXarcana Class
Letters	II:2 A-32–35	M. Díaz
Letterhead	II:2 A-33	M. Díaz
Lists	II:1 59, 72–72	A. Keller
	II:1 98–110	L. Price
	II:2 A-15	M. Díaz
Margins	II:2 A-19	M. Díaz
Matrices	II:2 A-30	M. Díaz
Memos	II:2 A-32–35	M. Díaz
Miscellaneous		
automatic printing of macro names	II:3 60–61	L. Price
avoiding "Argument of ⟨control sequence⟩ has an extra }."	II:2 50	M. Spivak
conditional evaluation of macros	II:2 50	M. Spivak
input-dependent macro redefinition	II:3 59–60	L. Price
\input within \if	II:2 50	M. Spivak
single tokens, identifying	II:2 52	M. Spivak
Multiplication	II:2 47	B. McKay

Nofill		
macros	II:1 59–60, 74–76	A. Keller
	II:2 A-16–18, 36	M. Díaz
program (SAIL)	II:1 87–93	L. Price, P. Milligan
program (Pascal)	II:1 94–97	L. Price, P. Milligan
program errata (SAIL and Pascal)	II:2 43–44	
Notes		
output to the writer on a separate file	II:1 60, 76, 85	A. Keller
printed at end of document	II:2 A-25	M. Díaz
Null string, testing for	II:1 60, 77	A. Keller
	II:2 51–52	M. Spivak
Numbering, page	II:1 57, 70–71	A. Keller
Output routines	II:1 57–58, 60–62, 71, 82–85	A. Keller
	II:2 A-18, 40	M. Díaz
Overlining	II:2 A-13	M. Díaz
Page numbering	II:1 57, 70–71	A. Keller
	II:2 A-18, 23	M. Díaz
Paragraphs		
beginning with large capital letters	II:1 60, 78	A. Keller
	II:2 A-16	M. Díaz
indented	II:1 58, 72	A. Keller
	II:2 A-13–15	M. Díaz
numbered, see Lists		
Parentheses, assorted sizes	II:2 A-11	M. Díaz
Pictures, plotting	II:2 48–49	B. McKay
Point, declaring font families of a particular ~ size	II:1 56–57, 65–66	A. Keller
	II:2 A-11	M. Díaz
Proofs	II:2 A-31–32	M. Díaz
Recursion	II:2 46–48	B. McKay
	II:2 53	M. Spivak
References	II:2 A-25	M. Díaz
Roman numerals, uppercase	II:1 120–121	P. Milligan, L. Price
Seating charts	III:1 39	R. Beeman
Spanish	II:2 A-12	M. Díaz
Strings		
testing for ~ equivalence	II:3 61	L. Price
testing for the null ~	II:1 60, 77	A. Keller
	II:2 51–51	M. Spivak
Syntax charts	II:3 39–56	M. Plass
Table of Contents	II:1 60, 62, 86	A. Keller
	II:1 111–118	L. Price
	II:2 A-27–28	M. Díaz
	II:3 24	B. Beeton
Tables	II:2 A-25–27	M. Díaz
Testing		
integral values	II:1 119–120	P. Milligan, L. Price
math-style (display, script or scriptscript)	II:2 46	B. McKay
for string equivalence	II:3 61	L. Price
for the null string	II:1 60,77	A. Keller
	II:2 51–52	M. Spivak
Theorems	II:2 A-31–32	M. Díaz
Top, baseline set to ~ of box	II:1 60, 77	A. Keller
TUGboat submissions	II:1 53–54	B. Beeton
	II:3 25	B. Beeton
Underlining	II:1 59, 73	A. Keller
	II:2 A-13	M. Díaz
Uppercase letters		
large ~ at beginning of paragraph	II:1 60, 78	A. Keller
	II:2 A-16	M. Díaz
Roman numerals	II:1 120–121	P. Milligan, L. Price

Verbatim			
mode	II:1	59–60, 74–76	A. Keller
. . . .	II:2	A-16–18, 36	M. Diaz
program (SAIL) . . .	II:1	87–93	L. Price, P. Milligan
program (Pascal) . . .	II:1	94–97	L. Price, P. Milligan
Vertical text	II:3	64	TEXarcana Class

* * * * * * * * * * *

DISPLAY OF A FONT IN TABLE FORM

Roger L. Beeman
Boeing Aerospace Company

`baselineskip` and `lineskip` are turned off to get them out of the way. `vsize` is increased to the size of my Versatec page. The output routine is redefined mostly to turn off the page numbering but `advancecount` is retained so that the page numbers displayed on the terminal will advance.

The character 0 from `cms10` is boxed so that its height and width will be available. `spike` defines an empty vbox which is used to assure that the horizontal rows are tall enough for the row number to fit without overfilling. `cell` is the basic box that holds one character, centered with a vertical rule on its right border. `label` uses the height of `box8` which may be different for each row and centers the octal tag rather than putting it on the same baseline as the rest of the row. The box width of 35pt is used to allay fears that the labels would not all turn out the same width and must be known later anyway. The `1em` of skip is inside the brackets and thus taken from `cms10`. `seprow` is used to add 2pts to the top and bottom of each cell.

`cellrow` saves the row of eight cells in `box8` so that label can use `ht8` for vertical centering. The spike is used to guarantee a minimum height. The height before boxing will be the maximum of this and the tallest character plus the 2pts from `seprow`. The boxing will cause a box of zero depth with the final height also including the maximum depth plus another 2pts from the second `seprow` and the height of the hrule.

This is probably the best place to point out what I really wanted was for the height above the highest character to equal the depth below the baseline. As it is, there is 2pt above the highest character and 2pt below the deepest. I probably wouldn't have given up except that `cmr30` was already pretty tight on the page and page breaking was not appetizing. Actually when it was working this well I was pretty relieved.

`lcol` labels the top, again in `cms10`. `chw`, `colw` and `setw` are used to find the maximum width of any character in the font. `getw` takes the maximum over the set of characters in the font, the width of the 0 used in labeling the columns, and `1em` in the font (maybe unnecessary) then sets the variable unit to $1\frac{3}{8}$ this value. The `1vu` is used as the width of each character cell.

Finally, `table` is defined to use the given character to define the font, set the font and build the table. The `hbox` has glue to center if possible but to left justify with right overfilling forgiven if necessary. The font name is included in `cms10`. The top label and the top rule for the font cell set are followed by the sixteen cellrows.

Editor's note: The two tables which follow were pasted up from Varian copy generated at the Math Society. A few changes were necessary: new letter codes were assigned to the two fonts because of conflicts with codes already assigned to preloaded fonts; `cmr28` *does not exist at the Society, so* `cmr30` *was substituted.*

We discovered after looking at the first output that this routine neatly illuminates probable errors in a couple of METAFONT *descriptions. In the* `cmr30` *table, row '000 has too much depth, and character '121, "Q", has no depth where one would have expected it. On checking the* METAFONT *descriptions, we found that the depth of the "Q" has disappeared (presumably accidentally—it was present in the original published description of the Computer Modern family), and that character '002, "Θ", has always been assigned a depth equal to that of a comma.*

cmathx

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*
'000	(	)	[	]	⌊	⌋	⌈	⌉
'010	{	}	⟨	⟩	|	‖	/	/
'020	(	)	(	)	[	]	⌊	⌋
'030	⌈	⌉	{	}	⟨	⟩	/	/
'040	(	)	[	]	⌊	⌋	⌈	⌉
'050	{	}	⟨	⟩	/			
'060	⎛	⎞	⎡	⎤	⎣	⎦	⎢	⎥
'070	⎧	⎫	⎩	⎭	⎨	⎬	⎪	
'100	⎝	⎠	⎜	⎟			⊔	⊔
'110	∮	∮	⊙	⊙	⊕	⊕	⊗	⊗
'120	Σ	Π	∫	⋃	⋂	⊎	⋀	⋁
'130	Σ	Π	∫	⋃	⋂	⊎	⋀	⋁
'140	⬭	○	○					
'150								
'160	√	√	√	√	⎷	|	┌	
'170	▸						G	

cmr30

	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
'010	Φ	Ψ	Ω	ı	ȷ	`	´	ˆ
'020	ˇ	˘	¯	¨	˜	→	˝	˚
'030	¸	-	⁀	ß	æ	œ	Æ	Œ
'040	ø	!	”	′	ft	%	&	’
'050	(	)	*	+	,	-	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	“	]	–	—
'140	‘	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	ff	fi	fl	ffi	ffl

Font Table Macros

```
\input basic
\baselineskip 0pt \lineskip 0pt
\vsize 7.3 in

\output{\vbox{\hrule width1em\vskip0.4in\page\vskip0.4in
\hrule width1em}\advcount0}

\save9\hbox{\sl 0}
\def\spike{\hbox to 0pt{\vbox to 1ht9{}}}
\def\cell#1{\hbox to 1vu{\hfill\char'#1\hfill}\vrule}
\def\label#1{\vbox to 1ht8{\vfill
  \hbox to 35pt{\hfill\sl '#10\hskip1em}\vfill}\vrule}

\def\seprow{\def\m{\hskip 1vu{}\vrule height 2pt}\hbox{\m\m\m\m\m\m\m\m}}
\def\cellrow#1{\save8\vbox{\seprow\hbox{\spike\!
               \cell{#10}\cell{#11}\cell{#12}\cell{#13}\!
               \cell{#14}\cell{#15}\cell{#16}\cell{#17}\!
               \hfill}\seprow\hrule}\hbox{\label{#1}\box8}}

\def\lcol#1{\hbox to 1vu{\hfill{\sl #1}\hfill}\hskip .4pt}
\def\chw#1{\hbox{\char'#1}}

\def\colw#1{\vbox{\chw{#10}
                  \chw{#11}
                  \chw{#12}
                  \chw{#13}
                  \chw{#14}
                  \chw{#15}
                  \chw{#16}
                  \chw{#17}}}

\def\setw#1{\vbox{\colw{#10}
                  \colw{#11}
                  \colw{#12}
                  \colw{#13}
                  \colw{#14}
                  \colw{#15}
                  \colw{#16}
                  \colw{#17}}}

\def\getw{\save0\vbox{\setw0\setw1\hbox to 1wd9{}
          \hbox to 1em{}}\varunit1.625wd0}

\def\table#1#2{\font #1=#2 \:#1\getw\null\vfill
\hbox to 6in{\hskip0pt plus1000cm\!
\vbox{\hbox{\hskip35pt{\sl #2}\hfill}
      \vskip20pt
      \hbox{\hskip35pt\lcol0\lcol1\lcol2\lcol3\lcol4\lcol5\lcol6\lcol7}
      \vskip 4 pt
      \hbox{\hskip35pt\vbox{\hrule width 8vu}\vbox{\hrule width 3.6pt}}
      \cellrow{00}\cellrow{01}\cellrow{02}\cellrow{03}
      \cellrow{04}\cellrow{05}\cellrow{06}\cellrow{07}
      \cellrow{10}\cellrow{11}\cellrow{12}\cellrow{13}
      \cellrow{14}\cellrow{15}\cellrow{16}\cellrow{17}}\!
\hskip0pt plus1000cm minus1000cm}\vfill\eject}

\table G{cmathx}                % \table A{cmathx}
\table N{cmr30}                 % \table B{cmr28}

\end
```

SEATING CHARTS

Roger L. Beeman
Boeing Aerospace Company

baselineskip and lineskip are set and a simple output format defined. Pat MacVicar–Whelan has the longest name and therefore controls the size of the large desks. Notice that I should have used the -- ligature.

The recurring theme through most of the macro definitions is the box from Exercise 21.3 of the TEX manual. I am not proud of the brute force approach but hopefully I will feel compelled to redeem myself by further contributions to TUGboat which will show I have progressed.

desk is a basic desk with a name in it. deskcom is the box for the com–line number. Negative glue is used to overlay into the large desk area on both sides. odeskcom is for the two guys who do not have to share their phones. Here the box is not balanced between two desks but set fully into the single desk. ndeskcom and sdeskcom reflect that the view is north looking, hence there are north and south desks. These two macros build the section between two desks, with the com–line box to the front of the desk whether north facing or south facing. name provides a box with hrules and a centered name.

ndeskpair and sdeskpair then make north fac ing and south facing pairs and ndeskrow and sdeskrow make rows of four desks. vdesk and hdesk are vestigial. vpdesk and hpdesk are the vertical and horizontal versions of desks with phones for the lower area. handesk is the horizontal desk which is annotated with the telephone number that gets answered as "*manager's name*'s office" instead of "Guidance and Navigation Technology".

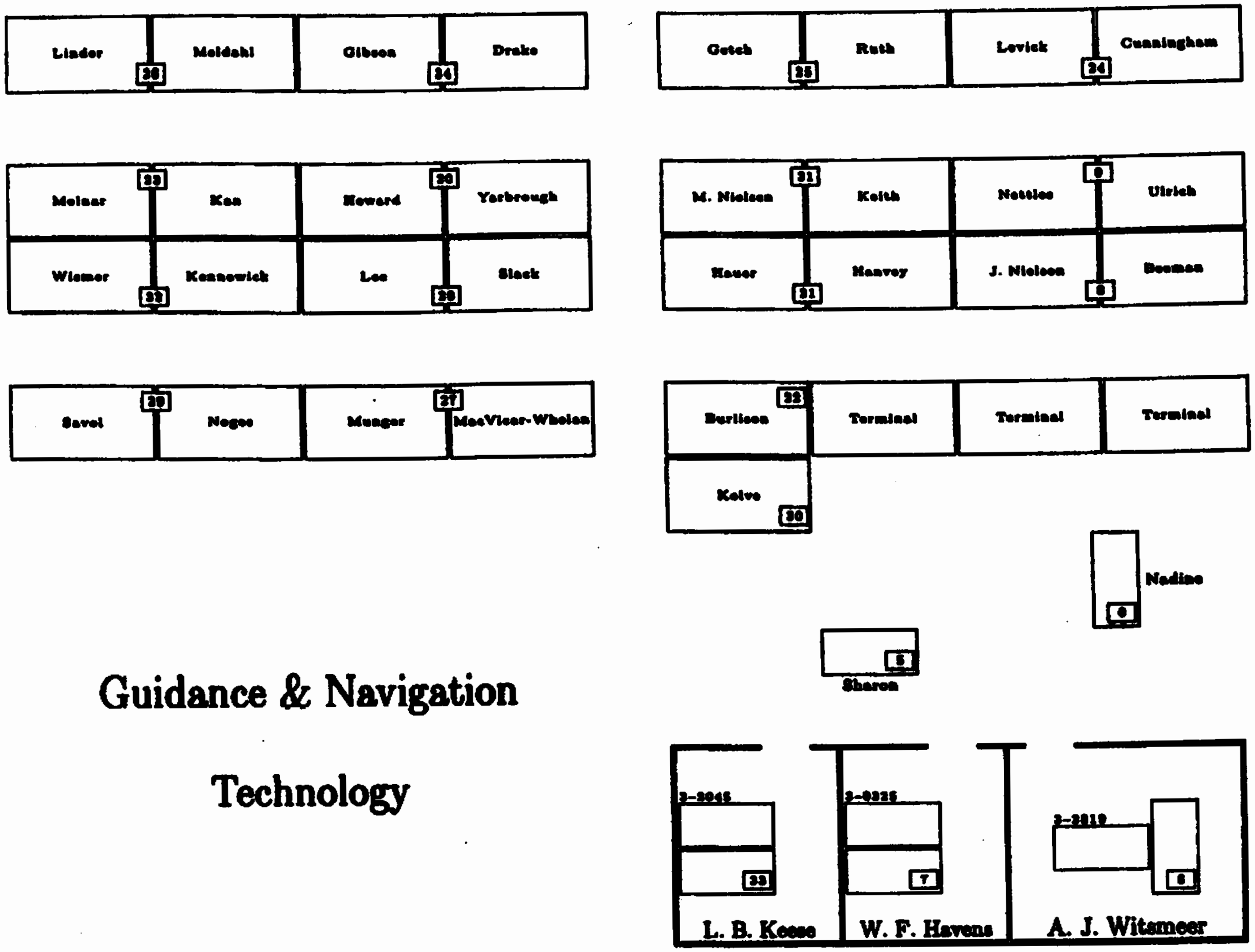
hwall and vwall do only the horizontal lines for horizontal and vertical wall sections. firstoffice and secondoffice do the offices for the two first level supervisors and the second level supervisor.

firstoffice haligns the top wall (with the door opening), the office interior and the bottom wall. The interior has the inside vertical walls around a left justified vbox. In the vbox are the annotated desk, the phone desk and the supervisor's name centered in a funny way. secondoffice is much the same except that a valign is used to position the desks differently.

Now the engineers' desks are put together straightforwardly with only a little fooling around to give Burlison and Kolve their phones and put Kolve's desk where it belongs without a visible ndeskrow to his left. The varunit is changed to scale the office layouts to fit on the same page. These were done seperately first with the units used representing measurements taken from a blueprint. The save9 is used to force evaluation of the nested alignments in the office area to allow the title to be aligned with the offices without exceeding nesting depth restrictions. halign and valign are used to put the secretaries' names by their desks.

This chart has proven much more useful than simple typewritten lists of com–line assignments and separate simple seating layouts. The secretaries also appreciate not having to type up new lists every couple weeks as personnel changes are made. One complaint I have received about the chart has been from a "south facer" who thinks that the north facing chart discriminates unfairly against his class. As a "north facer" myself I find it hard to sympathize.

Editor's note: A few minor changes were made in order to set this chart for TUGboat: two fonts were given new letter codes because of conflicts with codes already assigned to preloaded fonts, and cmr18 *(the largest font currently available on the Math Society's Alphatype) was substituted for* cmr22*; it was also necessary to substitute "\&\␣" for the ordinary &, since that character is by convention used as an alignment tab (see the errata list, page 21, regarding changes for TEX82). It should also be obvious that the chart was prepared separately and stripped in broadside: rotated fonts do not yet exist for the Alphatype.*



Guidance & Navigation

Technology

Seating Chart Macros

```
\input basic
\font b=cmr9
\font e=cmr6                      % \font c=cmr6
\font G=cmr18                     % \font e=cmr22
\curfont f
\baselineskip 0pt  \lineskip 0pt
 \output{\hbox to 8in{\hfill\page\hfill}}

\save0\hbox{MacVicar-Whelan}\varunit.024wd0

\null\vskip .15 in

\def\desk#1{\vbox{\hbox{\vrule\vbox to 22vu{
\hrule\vfill\hbox to 43.2vu{\hfill#1\hfill}\vfill\hrule}\vrule\hskip1pt}
\vskip1pt}}

\def\deskcom#1{\vbox{
\hbox to 1.8pt{\hss\vrule\vbox to 5.5vu{\hrule\vfill
\hbox to 8vu{\hfill{\curfont f #1}\hfill}\vfill\hrule}\vrule\hss}}}

\def\odeskcom#1{\vbox{
\hbox to 0pt{\hss\vrule\vbox to 5.5vu{\hrule\vfill
\hbox to 8vu{\hfill{\curfont f #1}\hfill}\vfill\hrule}\vrule\hskip2pt}}}

\def\ndeskcom#1{\vbox{\hbox{\vrule\vbox to 14.5vu{\vfill}\hskip1pt\vrule}
\deskcom{#1}\hbox{\vrule\vbox to 2vu{\vfill}\hskip1pt\vrule}}}

\def\sdeskcom#1{\vbox{\hbox{\vrule\vbox to 2vu{\vfill}\hskip1pt\vrule}
\deskcom{#1}\hbox{\vrule\vbox to 14.5vu{\vfill}\hskip1pt\vrule}}}

\def\name#1{\vbox to 22vu{
\hrule\vfill\hbox to 43.2vu{\hfill#1\hfill}\vfi!l\hrule}}

\def\ndeskpair#1#2#3{\vbox{\hbox{\vrule\!
\name{#1}\ndeskcom{#2}\name{#3}\vrule\hskip1pt}\vskip1pt}}

\def\sdeskpair#1#2#3{\vbox{\hbox{\vrule\!
\name{#1}\sdeskcom{#2}\name{#3}\vrule\hskip1pt}\vskip1pt}}

\def\ndeskrow#1#2#3#4#5#6{\hbox{\ndeskpair{#1}{#2}{#3}\ndeskpair{#4}{#5}{#6}}}

\def\sdeskrow#1#2#3#4#5#6{\hbox{\sdeskpair{#1}{#2}{#3}\sdeskpair{#4}{#5}{#6}}}
\def\vdesk{\hbox{\vrule\vbox to 19vu{
\hrule\vfill\hbox to 9vu{\hfill}\vfill\hrule}\vrule}}

\def\hdesk{\hbox{\vrule\vbox to 9vu{
\hrule\vfill\hbox to 19vu{\hfill}\vfill\hrule}\vrule}}

\def\vpdesk#1{\hbox{\valign{##\cr
\hbox{\vrule\vbox to 19vu{\hrule\vfill\hbox to 9vu{\hfill}
\vfill\hrule}\vrule}\cr
\vfill\hbox to 0pt{\hss\vrule\vbox to 3.6667vu{
\hrule\vfill\hbox to 5.3333vu{\hfill{\curfont f #1}\hfill}\vfill\hrule}\!
\vrule\hskip.7vu}\vskip1vu\cr}}}

\def\hpdesk#1{\hbox{\valign{##\cr
\hbox{\vrule\vbox to 9vu{\hrule\vfill\hbox to 19vu{\hfill}
\vfill\hrule}\vrule}\cr
\vfill\hbox to 0pt{\hss\vrule\vbox to 3.6667vu{
\hrule\vfill\hbox to 5.3333vu{\hfill{\curfont f #1}\hfill}\vfill\hrule}\!
\vrule\hskip.7vu}\vskip1vu\cr}}}

\def\handesk#1{\hbox{\vbox{\vbox to 0pt{\vskip0pt minus 1000cm
\hbox{\curfont f {#1}}\vskip1.2pt}
\hbox{\vrule\vbox to 9vu{\hrule\vfill\hbox to 19vu{\hfill}
\vfill\hrule}\vrule}}}}

\def\hwall#1{\vbox{\hrule\hbox{\hskip#1}\vskip.8pt\hrule}}

\def\vwall{\hrule\vfill\hbox{\hskip.8pt}\hrule}

\def\firstoffice#1#2#3{\halign{##\cr\!
\hwall{17.5vu}\vrule\hfill\vrule\hwall{5.5vu}\cr\!
\vrule\vbox to 39vu{\vskip11vu
\hbox{\hskip1pt\vbox{\handesk{#3}\vskip1pt\hpdesk{#2}}}
\vfill\hbox{\hskip 16.5vu\hbox to 0pt{\hss{\curfont b #1}\hss}}
\vskip2pt}\hfill\vrule\cr\!
\hwall{33vu}\hwall{.8pt}\cr}}

\def\secondoffice#1#2#3{\halign{##\cr\!
\hwall{3.5vu}\vrule\hfill\vrule\hwall{33.5vu}\cr\!
\vrule\vbox to 39vu{\vskip11vu
\hbox{\hskip 23.5vu\hbox to 0pt{\hss\!
\valign{##\cr
\vfill\handesk{#3}\vfill\cr\hbox{\hskip1pt\vpdesk{#2}}\cr}\hss}}
\vfill\hbox{\hskip 23.5vu\hbox to 0pt{\hss{\rm #1}\hss}}
\vskip2pt}\hfill\vrule\cr\!
\hwall{47vu}\hwall{.8pt}\cr}}

\hbox{\ndeskrow{Linder}{26}{Meldahl}{Gibson}{34}{Drake}\hskip.4in\!
\ndeskrow{Getch}{25}{Ruth}{Levick}{24}{Cunningham}}
\vskip.4in
\hbox{\sdeskrow{Molnar}{23}{Kan}{Howard}{20}{Yarbrough}\hskip.4in\!
\sdeskrow{M. Nielsen}{31}{Keith}{Nettles}{9}{Ulrich}}
\hbox{\ndeskrow{Wismer}{32}{Kennewick}{Lee}{28}{Slack}\hskip.4in\!
\ndeskrow{Hauer}{21}{Hanvey}{J. Nielson}{8}{Beeman}}
\vskip.4in
\hbox{\sdeskrow{Savol}{29}{Noges}{Munger}{27}{MacVicar-Whelan}\hskip.4in\!
\hbox{\desk{Burlison}\vbox{\vskip2vu\odeskcom{22}
\vskip14.5vu\vskip1pt}\desk{Terminal}\desk{Terminal}\desk{Terminal}}}
\hbox{\save0\hbox{\ndeskrow{}{}{}{}{}{}\hskip.4in}\hskip1wd0\!
\hbox{\desk{Kolve}\vbox{\vskip14.5vu\odeskcom{30}\vskip2vu\vskip1pt}}}

\varunit1.5vu

\save9\vbox{\hbox{\hskip86vu\valign{#\cr\vpdesk{0}\cr\hbox{\hskip2.6pt}\cr
\vfill\hbox{\curfont e Nadine}\vfill\cr}}
\hbox{\hskip31vu\vbox{\halign{#\cr\!
\hpdesk{5}\cr\noalign{\vskip2pt}\hfill{\curfont e Sharon}\hfill\cr}}}
\vskip11vu
\hbox{\vrule\valign{#\cr
\vwall\cr\firstoffice{L. B. Keese}{33}{3--2045}\cr\vwall\cr
         \firstoffice{W. F. Havens}{7}{3--0325}\cr\vwall\cr
         \secondoffice{A. J. Witsmeer}{6}{3--2819}\cr\vwall\cr}\vrule}}

\hbox{\valign{#\cr
\vfill\vbox{\halign{#\cr\hbox{\hskip1wd0}\cr\hfill\!
{\curfont G Guidance \&\ Navigation}\hfill\hskip.4in\cr\noalign{\vskip25pt}\!
\hfill{\curfont G Technology}\hfill\hskip.4in\cr}}\vfill\cr
\box9\cr}}

\vfill
\end
```

* * * * * * * * * * *

Problems

* * * * * * * * * * *

PROBLEMS FROM THE TEXARCANA CLASS: ANSWER TO THE CHALLENGE PROBLEM

The problem is, produce this:

•If I have all the eloquence of men or of angels, but speak without love, I am 1
simply a gong booming or a cymbal clashing. •If I have the gift of prophecy, 2
understanding all the mysteries there are, and knowing everything, and if I have
faith in all its fulness, to move mountains, but without love, then I am nothing
at all. •If I give away all that I possess, piece by piece, and if I even let them 3
take my body to burn it, but am without love, it will do me no good whatever.
•Love is always patient and kind; it is never jealous; love is never boastful 4
or conceited; •it is never rude or selfish; it does not take offence, and is not 5
resentful. •Love takes no pleasure in other people's sins but delights in the truth; 6
•it is always ready to excuse, to trust, to hope, and to endure whatever comes. 7
•Love does not come to an end. But if there are gifts of prophecy, the time will 8
come when they must fail; or the gift of languages, it will not continue for ever;

(and a second page)

and knowledge—for this, too, the time will come when it must fail. •For our 9
knowledge is imperfect and our prophesying is imperfect; •but once perfection 10
comes, all imperfect things will disappear. •When I was a child, I used to talk 11
like a child, and think like a child, and argue like a child, but now I am a man,
all childish ways are put behind me. •Now we are seeing a dim reflection in a 12
mirror; but then we shall be seeing face to face. The knowledge that I have now
is imperfect; but then I shall know as fully as I am known.
•In short, there are three things that last: faith, hope and love; and the greatest 13
of these is love.

from a data file, `love.tex`, that looks like this:

```
@ If I have all the eloquence of men or of
    angels, but speak without love, I am simply a gong booming
    or a cymbal clashing.
@ If I have the gift of prophecy, understanding all the mysteries
    ...
    imperfect; but then I shall know as fully as I am known.

@ In short, there are three things that last: faith, hope and
    love; and the greatest of these is love.
```

Here is Don Knuth's solution.

```
\input basic
\def\pagesize{1.8in}

\vsize 8.5pt
\topbaseline 8.5pt
\maxdepth 3.5pt

\output{\save0\vbox{\unbox0\hbox{\lower3.5pt\hbox{}\page}}
\ifdimen 1ht0<\pagesize{}
\else{\vbox to \pagesize{\box0\vss}\advcount0}
}
\def\flushpage{\par\eject\ifdimen 1ht0=0pt{}
\else{\vsize\pagesize\vbox to \pagesize{}\eject\vsize 8.5pt} }

\topsep{\vskip -6pt minus 10pt}
\topskip 0pt minus -5pt
\def\margno{\topinsert{\vskip0pt minus 5pt
\rjustline{\vbox to 8pt{}\spose{\hskip 5pt\:d\count1}\advcount1}}\!
}

\chcode'100=13 \def@{\ifvmode{\ \unskip}
\else{\unskip\hskip 1.5em plus .1em minus .1em}
\hbox to 0pt{\hss$\bullet$}\margno}

\setcount1 1
\input love
\end
```

* * * * * * * * * * * *

Late-Breaking News

* * * * * * * * * * * *

TUG FINANCIAL REPORTS

Samuel B. Whidden, Treasurer

Two reports follow—the TUG treasurer's report for 1981 and a tentative budget for 1982. They reside here in the late-breaking news section because we delayed preparing them until we had nearly-final 1981 figures from the AMS accounting system (the accounts are still unaudited, so the final figures may still be slightly different).

These reports present a gloomier-than-expected picture of TUG finances. They show that we ended 1981 with a deficit of over $8,600. In part, this is because our books are now being kept on an 'accrual' rather than a 'cash' basis, so all income and expense relating to 1982 appear in the 1982 budget even if money was actually received or spent in 1981. Accrual accounting makes the 1981 deficit about $4,900 greater than it would have appeared on a cash basis. Another $3,100 in 1981 expenses appear in this report which were recorded too late to be reported to us by the time of the Cincinnati meeting.

Also included in this report are about $5,800 in clerical costs which TUG believed would be absorbed by AMS, but which were not (this was a misunderstanding on our part—AMS did absorb more than $17,000 of non-clerical TUG expenses and wrote off the $419 negative balance, left over from 1980 clerical charges, with which we started 1981). AMS overhead charges account for another $3,800 (overhead of 18.84% has been added to each expense item). The net deficit for 1981 amounts to $8,663. AMS continues to carry this deficit for us, but wants to be repaid eventually.

The picture for 1982 is somewhat more encouraging in that the income-producing and cost-saving measures authorized at the Cincinnati meeting should cut the operating deficit at least in half (even though AMS will be absorbing almost none of our costs this year), but we will still be left with a cumulative deficit of more than $12,000 at the end of 1982. We have had to make a number of assumptions, of course, in projecting 1982 income and expense (see the notes to the 1982 budget), but we've tried to make our estimates realistic, rather than optimistic or pessimistic. Actual experience will certainly be different. An important part of the meeting at Stanford this July will be the Steering Committee's review of those assumptions and estimates, and of TUG's financial position then.

The AMS trustees, at their meetings this Spring, will review their policy of supporting TUG deficits until we can repay them. Whether they will continue this policy or revise it is unknown at present.

* * * * * * * * * * * *

TUG TREASURER'S REPORT

December 31, 1981

Ending balance, December 31, 1980:			$(419)
Adjusting entry, January 1, 1981:			419
Income: 1981 Membership[1]		$ 6,125	
Tape leasing/sales		1,250	
Stanford Workshop[2]		7,695	15,070
Expenses[3]:			
TUGboat Vol. 2 (1981) 3 Issues			
Printing	$6,182		
Postage	2,230		
Clerical	3,926	$12,338	
Reprinting TUGboat:			
Vol. 1, No. 1: 300 copies		231	
Microfiche TUGboat:			
Vol. 1, No. 1 and			
Vol. 2, No. 1		130	
Questionnaire/membership renewal			
Printing	$ 119		
Mailing	178	297	
Miscellaneous postage, express charges		581	
Knuth TEX lectures, videotape duplication		1,009	
Steering Committee luncheon meeting, San Francisco, January '81		202	
Stanford Workshop expenses[2]		497	
Supplies/photocopies		379	
Clerical support[4]		3,684	
Telephone tolls		115	
Support for Stanford TEX coordinator[5]		4,270	$(23,733)
Ending Balance:			$(8,663)

Notes:

All expense figures include an AMS overhead charge of 18.84%.

1. 1981 memberships numbered 634, 22 of which were complimentary.
2. The TEX Implementors' Workshop held at Stanford, May 14–15, 1981, was attended by 92 participants.

3. Includes clerical/administrative services associated with the production of TUGboat to include maintaining the mailing list; not included in these figures are costs for services provided by AMS professional staff, including programming, reviewing and editing and answering telephone inquiries amounting to $14,376, plus 18.84% overhead charge.
4. Clerical services for creation and maintenance of TUG membership data base plus various clerical services not associated with production of TUGboat.
5. Professor Arthur Samuel is acting for Luis Trabb-Pardo as TeX coordinator, answering questions, distributing tapes, and fixing bugs in the TeX source code. Luis has asked, and the finance committee has agreed, that TUG contribute to Professor Samuel's support during 1981. The amount given Stanford was $3,600; AMS overhead of 18.84% increases the cost to TUG to $4,270.

Respectfully submitted,
Samuel B. Whidden, Treasurer

* * * * * * * * * * *

TUG BUDGET – 1982

Income:		
Membership/Publications		
1981 Membership,[1] 50@$10	$ 500	
1981 Back volume sales,[1] 35@$30	1,050	
1982 Membership,[2] 600@$15	9,000	
1982 Library subscriptions,[3] 30@$15	450	
1982 Foreign postage option,[2] 50@$12	600	
Supplements[4,*]	500	$12,100
Meetings		
Cincinnati, 1/82	$ 4,500	
Stanford, 7/82	3,000	7,500
Institutional Membership		
Educational,* 15@$250	$ 3,750	
Non-educational,* 10@$500	5,000	8,750
Other		
Videotape sales/rental	$ 3,000	
Advertising and mailing list sales*	1,000	
Royalties (TeX manual)*	500	4,500
Total income		$32,850

Expenses:		
TUGboat (2 issues)[5]		
Printing	$ 2,830	
Postage	700	
Editorial services	4,920	
Clerical services	2,950	
Computer expense	3,310	$14,710
Meetings		
Cincinnati, 1/82[6]	$ 2,950	
Stanford, 7/82	1,770	4,720
Other		
Supplements[4]	$ 350	
TeX distribution support (Stanford)[7]	8,500	
ANSI meetings[8]	1,180	
Legal and tax consulting	1,180	
Advertising TUG membership & TUGboat[9]	1,770	
General mailings	710	
Subsidies[10]	1,180	
Miscellaneous[11]	2,360	17,230
Total expenses		$ 36,660

Budget summary:	
Balance forward 1/1/82	$(8,663)
Total income	32,850
Total expenses	(36,660)
Estimated balance 12/31/82	$(12,473)

Notes:

* These are categories of income which have not been previously budgeted and, therefore, represent our best estimate.

All expense figures have budgeted an AMS overhead charge of 18%.

1. 1981 memberships will be accepted through April 30, 1982, at $10 rate; thereafter, individual issues will be sold for $10 each (or $30 per volume/3 issues).
2. 310 1982 memberships @$15 each were received before December 31, 1981, for which income of $4,650 is recognized in this budget. Since January 1, an additional 117 1982 paid memberships have been received. Memberships are still coming in at a steady rate. We are projecting a total of 600, of which 173 are yet to be received. The two TUGboat issues to be published this year, plus the trade advertising we have been charged with doing, should yield at least this number. 40 members have subscribed to the $12 foreign air mail postage option.

3. Libraries may now subscribe to TUGboat without applying for individual membership.
4. Lengthy descriptions of macro packages will be available for purchase separately.
5. Editorial services include programming, reviewing and editing; clerical services include maintaining the data base and mailing list, and other administrative duties.
6. $1,500 has been allocated to partially cover expenses for participation by Don Knuth, Luis Trabb-Pardo, David Fuchs, Ignacio Zabala and Arthur Samuel in the TUG Cincinnati meeting, January 11–12, 1982. The balance is available to cover other expenses associated with the meeting.
7. Allocation to Stanford primarily for Professor Arthur Samuel, who is acting as TEX coordinator, answering questions, distributing tapes, and fixing bugs in the TEX source code.
8. The Steering Committee authorized attendance by Lynne Price at one meeting of ANSI X3J6. Budget would increase accordingly if attendance at additional meetings is authorized.
9. Costs for advertising TUG membership in trade publications.
10. Money available to Finance Committee to subsidize travel and membership fees for individuals when appropriate.
11. Postage/express charges, telephone tolls and supplies, plus programmer and clerical services not associated with production of TUGboat.

Respectfully submitted,
Samuel B. Whidden, Treasurer

TUGBOAT

VOLUME 3, NUMBER 1

Contents

March 1982

Addresses of Officers, Authors and Others. 2

Official Announcements. 3

General Delivery

Barbara Beeton. *Statement of Editorial Policy* 3

Robert Morris. *Report on the TUG Steering Committee Meeting* 4

Samuel B. Whidden. *TUG Financial Reports* *see Late-Breaking News*

Program, TUG Winter Meeting, January 11–12, 1982, Cincinnati, Ohio 5

Attendees, TUG Winter Meeting, January 11–12, 1982, Cincinnati, Ohio 6

Robert McGaffey and Keith Penny. *Open letter to TUG* 7

Lynne A. Price. *Report on the ANSI X3J6 Meeting* 10

Software

Donald Knuth. *Fixed-Point Glue Setting – An Example of* WEB 10

Site Reports

CDC Cyber

Michael J. Frisch. *CDC TEX* 28

Hewlett-Packard HP3000

Lance Carnes. *HP3000 Site Report* 28

IBM 370 and related architectures

G. Canzii, D. Lucarella, and A. Pilenga. *TEX at University of Milan* 29

L. Aiello and S. Pavan. *TEX News from Pisa* 30

VAX/VMS

Monte C. Nichols. *VAX/VMS Site Report* 32

"small" TEX

Lance Carnes. *Editor's Introduction* 32

Warnings & Limitations

Barbara Beeton. *Another Hangup* 33

Macros

Lynne Price. *Editor's Introduction* 33

TUGboat Macro Index 33

Roger L. Beeman. *Display of a Font in Table Form* 35

Roger L. Beeman. *Seating Charts* 39

Problems

Problems from the TEXarcana Class: Answer to the Challenge Problem 43

Late-Breaking News

Samuel B. Whidden. *TUG Financial Reports* 45

TUG Treasurer's Report 45

1982 TUG Budget 46