

PASCAL USER'S GROUP

USER'S
GROUP

PASCAL NEWSLETTER

NUMBER 7

COMMUNICATIONS ABOUT THE PROGRAMMING LANGUAGE PASCAL BY PASCALERS

FEBRUARY, 1977

TABLE OF CONTENTS

#			#
*			*
#			#
*			*
#	0	POLICY	#
*	1	EDITOR'S CONTRIBUTION	*
#	3	HERE AND THERE WITH PASCAL	#
*	3	News	*
#	4	Conferences	#
*	4	Books	*
#	4	Roster	#
*	8	ARTICLES	*
#	8	"Life, Liberty and the Pursuit of Unformatted Input"	#
*		- D. W. Barron and J. M. Mullins	*
#	9	"Pascal Printer Plotter"	#
*		- Herbert Rubenstein	*
#	17	"Yet Another Look at Code Generation for Pascal	#
*		on CDC 6000 and Cyber Machines"	*
#		- Lawrence A. Liddiard	#
*	24	OPEN FORUM FOR MEMBERS	*
#	27	IMPLEMENTATION NOTES	#
*	27	General Information	*
#	27	Checklist	#
*	27	Portable Pascals	*
#	29	Software Writing Tools	#
*	29	Compilers	*
#	45	ALL PURPOSE COUPON	#
*			*
#			#
*			*
#			#
*			*

PASCAL USER'S GROUP POLICIES

Purposes - are to promote the use of the programming language Pascal as well as the ideas behind Pascal. Pascal is a practical, general purpose language with a small and systematic structure being used for:

- * teaching programming concepts
- * developing reliable "production" software
- * implementing software efficiently on today's machines
- * writing portable software

Membership - is open to anyone: particularly the Pascal user, teacher, maintainer, implementor, distributor, or just plain fan. Institutional memberships, especially libraries, are encouraged. Membership is per academic year ending June 30. Anyone joining for a particular year will receive all 4 quarterly issues of Pascal Newsletter for that year. (In other words, back issues are sent automatically.) First time members receive a receipt for membership; renewers do not to save PUG postage.

Cost of membership per academic year is \$4 and may be sent to:
Pascal User's Group/ %Andy Mickel/University Computer Center/227 Exp Engr/
University of Minnesota/Minneapolis, MN 55455 USA/ phone: (612) 376-7290

In the United Kingdom, send £2.50 to:
Pascal Users' Group/ %Judy Mullins/Mathematics Department/The University/
SOUTHAMPTON/S09 5NH/United Kingdom/ (telephone 0703-559122 x2387)

PASCAL NEWSLETTER POLICIES

The Pascal Newsletter is the official but informal publication of the User's Group. It is produced quarterly (usually September, November, February, and May). A complete membership list is printed in the November issue. Single back issues are available for \$1 each. Out of print: #s 1,2,3
#4 available from George Richmond/Computing Center/U of Colorado/Boulder/80309

The contribution by PUG members of ideas, queries, articles, letters, and opinions for the Newsletter is important. Articles and notices concern: Pascal philosophy, the use of Pascal as a teaching tool, uses of Pascal at different computer installations, portable (applications) program exchange, how to promote Pascal usage, and important events (meetings, publications, etc.).

Implementation information for the programming language Pascal on different computer systems is provided in the Newsletter out of the necessity to spread the use of Pascal. This includes contacts for maintainers, documentors, and distributors of a given implementation as well as where to send bug reports. Both qualitative and quantitative descriptions for a given implementation are publicized. Proposed extensions to Standard Pascal for users of a given implementation are aired. Announcements are made of the availability of new software writing tools for a Pascal environment.

Miscellaneous features include bibliographies, questionnaires, and membership lists. Editor's notes are in Pascal style comments (**).

WRITTEN INFORMATION FOR THE Newsletter IS EASIER TO PRINT IF YOU TYPE ALL MATERIAL 1/2 OR DOUBLE SPACED SO THAT IT IS IN "CAMERA-READY" AND "PHOTO-REDUCIBLE" FORM FOR THE PRINTER. REMEMBER, ALL LETTERS TO US WILL BE PRINTED IN THE Newsletter UNLESS THEY CONTAIN A REQUEST TO THE CONTRARY. AN OVERRIDING GUIDE SEEN IN AN OLD MAD MAGAZINE APPLIES: "all the news that fits, we print!"

- Andy Mickel, editor, John P. Strait, associate editor, December 29, 1976.

POLICY



UNIVERSITY OF MINNESOTA
TWIN CITIES

University Computer Center
227 Experimental Engineering Building
Minneapolis, Minnesota 55455

(612) 376-7290

PART I - Promoting Pascal Usage

This concerns large computer centers with many users who can be thought of as consumers of computer software. If we consider only the users (including the computer center staff) who write programs in a programming language (rather than, say, use a package such as SPSS - Statistical Package for the Social Sciences) then the problem comes down to converting users from established and less desirable languages such as FORTRAN, ALGOL-60, BASIC, and PL/I to Pascal.

William Waite's valuable guest editorial in Software, Practice and Experience Vol. 3, pp. 195-6, provides many of the guidelines. He states that the support available for a language is more important than the features in the language itself. He likens languages to "species which inhabit certain ecological niches." The analogy has been quite useful in identifying ways of promoting Pascal at the University of Minnesota (see also Editor's Contribution Pascal Newsletter #5). Waite goes on to say that only when a more efficient competitor for a given language's life support systems is introduced into the ecosystem will the given language be ousted. At many universities, then, Fortran is a very healthy species which occupies, according to Waite, "the niche created by 'scientific' computation" (perhaps numerical computation is a better term).

The specific checklist given by Waite includes:

- 1) Is a sufficient library of standard procedures available?
- 2) Can the relevant ones be extracted automatically?
- 3) Is there easy interface with computer system utilities such as the file manager and the sort/merge package?
- 4) Is there easy interface for assembler if performance measurement shows critical procedures which are not adequately optimized by the translator?
- 5) Will consultation and programming assistance be available if I have problems with the language?
- 6) Is there sufficient program preparation equipment (interactive terminals, keypunches) with the proper character sets?
- 7) Will the computer system provide reasonable turnaround for programs written in the language?

EDITOR'S CONTRIBUTION

If we take an "advocacy" position with respect to Pascal in order to promote its use, these and other aspects come into play. At the University of Minnesota, the following proved very useful since we began using Pascal in 1972.

- 1) The prevailing view that a language processor should be given support only proportional to its usage cannot be tolerated. To give a language processor a fair chance, it must be actively promoted for awhile and then its acceptance evaluated. The intrinsic merit of the language and its processor should be the determining factors.
- 2) Proper user documentation helps promote the language. Pascal starts with Pascal User Manual and Report. In addition a local computer center should provide two documents: A) a double-sided one sheet handout describing the local Pascal facilities to satisfy the numerous requests for information made by walk-in users. On this sheet should be a date, place, author, and a description of the purpose of the sheet. The installation's Pascal facilities include references for more information (including P,UM&R and the other document described below); a description of the compiler or interpreter, its origin, reliability, commonly used options, and how to use the system in both batch and interactive modes (this includes command sequences and a description of the form of compile-time and run-time messages); and a small character set table if substitutions need to be made. B) a larger (20-30 page document) both in printed copy and in machine retrievable form which includes an introduction describing the scope of the document; information about the programming language Pascal, its history, uses, implementations, general and short description of its semantics and syntax; a history of the particular implementations (Pascal compilers) the computer center is running (current features and future developments); a description of the implementation, how it works, the specific definition of the sizes of scalar types, predefined (non-standard) identifiers, compiler options and switches, differences between this implementation and the standard; how to run programs under the implementation both in interactive and batch modes, program preparation, character sets, the commands to invoke the compiler or interpreter; guidelines, hints and cautions for effective usage, error messages, how to use software writing tools such as the cross-referencer, prettyprinter, source language editor, etc.; a detailed annotated list of references: introductory texts, reference manuals, books on applications, and sources of current information.
- 3) Enough people must be available for helping users with problems in their programs.
- 4) Publicity for Pascal to keep it constantly in the eye of the computer center user community: 2-3 week short courses in the language; articles in the computer center newsletter promoting the language as well as announcing planned changes in versions;

EDITOR'S CONTRIBUTION

living, "useful", well written, and simple example programs to show the language at its best: (e.g. a fancy calendar program - what better way to get people's interest?).

- 5) When converting persons remember that: don't wastetime converting Fortran and assembly language programmers who are overly concerned with machine efficiency. They will persist in their habits. Pascal's strengths lie in reducing the number of runs one has to make on program development, and because Pascal is nearly as efficient in terms of machine time as Fortran, less actual computer time is used; new programmers are the best bet, the computer science department can made a great contribution by teaching the language to new programmers and using it in other parts of the curriculum; urge people to write new programs in the language rather than getting them to convert old programs - although the latter may produce converts astounded at better solutions arrived at because they were able to think more clearly and conceive of a better algorighm in a systematic language (Pascal).

After bringing Pascal from nowhere to third out of 20 languages in four years we feel that support for Pascal is sufficient to survive and "ecological counterattack" and will continue to erode Fortran's base of users as we satisfy more of William Waite's principles - particularly in the area of libraries of procedures.

PART II - Pascal and Standards

There has not been time to receive the reaction to the proposals which appeared in Newsletter #6. Formal standardization of Pascal as it is now (by an official standards organization such as ISO or ANSI which could then have economic enforcement in the marketplace) is pretty straightforward. Changing Pascal is certainly a political problem, and even deciding how to pick a committee and when and where it could meet may prove to be overwhelming. A lot of issues regarding specific changes are not clear cut. We are a loose union, not a tight band, of devotees.

What we should concentrate on is conventionalizing the few recurring extensions in the various implementations of Pascal. We can use the Newsletter for that. I cannot overemphasize my conviction that much, careful consideration was given to what features were left out of Pascal as it has evolved. We must always go back to the design goals of compactness, vehicle for portability, vehicle for teaching systematic programming, and a tool to write efficient production programs. We should not use Pascal for purposes it was not intended (such as writing an operating system). One should not misuse or break any tool. Note that other languages have been designed for those tasks (in the case of operating systems with the need to express concurrent

processes we have Brinch Hansen's Concurrent Pascal, Hoare's SIMONE, and a rumor about Wirth's MODULA).

I hope my editorial in PUGN#6 did not seem too confused - I was trying to be compromising and all-encompassing and I still lack a lot of answers.

One final note: the wholeSALE bending of Pascal to make it conform to conventions of Burroughs ALGOL (as described in the report: "Burroughs Pascal: Some Implementor's Thoughts) is alarming. Why have a different language (Pascal) available which can bridge Burroughs users to software written on other machines if one adopts so many features from an existing Burrough's language: Burroughs Extended ALGOL? One might as well stick to Burroughs ALGOL. At best it's PascALGOL! Why get so upset? Implementors are not operating in a vacuum; they affect all of us on the issue of standards and portability.

PART III - PUG and Pascal Newsletter

PUG now has 598 members in 24 countries and 44 states. We have been growing steadily at the rate of 60 members/month since we started. By June, then, we should have almost 1000 members. This will cause us financial anguish in the form of growing pains. So the result is very poor service regarding back issues. Sorry. It's hard to plan ahead with the small budget we have.

Speaking of slow distribution, as I write this I'm sure some of you have not yet received #6. This will make #8 longer when the reaction arrives. #5 and #6 were big and were mailed overseas by air so that we could get the Pascal movement back on track. We also printed nearly everything that came to our attention. This satisfied the individual urgent questions we had been receiving about newsletters and implementations. This did cost us money and that is why #7 is small and will not go by air overseas. We are happy to report that things seem to be much improved as the information has gotten out via the newsletter. We have tried to take an advocacy position with the goal of furthering Pascal. We hope that this hasn't offended anyone.

The bright area is the UK distribution center which is getting lots of members and has eased the overseas distribution load for us.

Thanks for all the compliments regarding the newsletter. We shall try to "keep up the good work." But we do need the help requested for handling some departments of the newsletter and functions of the User's Group written in PUGN#6.



December 29, 1976

NEWS (ALPHABETICAL BY LAST NAME)

Jim Fontana, Control Data Corporation, 3519 W. Warner Ave., Santa Ana, CA 92704 (PUG member): "... I would appreciate hearing from any PUG members who know where tapes and listings for any version of Concurrent Pascal and the Solo Operating System are available." (*12/20/76*)

Bill Hopkins, Dept. of Comp. and Info. Sci., University of Pennsylvania, Philadelphia, PA 19174 (PUG member): "...I am at the moment Pascal-less. Do you have any Univac (nee RCA) Spectra 70 installations on the mailing list? We're going to a 9070 Univac in the Christmas break but it will be software compatible - still Univac VMOS. I understand that the Stony Brook 360 compiler was being converted to VMOS at Georgia State, but I haven't been able to raise anyone there. Would appreciate any pointers." (*10/2/76*)

Pei Hsia, Computer Science Program, The University of Alabama, Huntsville, P.O. Box 1247, Huntsville, AL 35807 (PUG member): "...We currently have a Univac 1100/10 system and are planning to implement a Pascal in order to teach some basic programming concepts of it." (*12/6/76*)

Olivier Lecarme, I.M.A.N., Universite de Nice, Parc Valrose, Nice Cedex 06034, France (PUG member): We received the 173 page proceedings from a "Pascal Days" conference held in France during June 5th and 6th, 1975. Olivier co-organized the conference which discussed implementation, development, and applications in 7 working sessions. C. Girault was the other co-organizer. He is at the Institut de Programmation, Universite Paris VI. Other participants were: J. Cea, D. Thibault, J.L. Bouchenez, B. Lohro, C. Girault, M. de la Croix, M. Galinier, J.L. Pouzin, J. Farre, M. Gauthier, A. Tisserant, G. Terrine, P. Maurice, J.P. Partouche, G. Tassart, D. Thibault, B. Lang, D. Fortier, D. Gurtner, J.L. Nebut, R. Rousseau, G. Terrine, Nguyen Van Lu, B. Robinet, M. Dupras, C. Precetti, and J. Bezivin. (*12/13/76*)

John L. Norstad, Northwestern University, Vogelback Computing Center, 2129 Sheridan Rd., Evanston, IL 60201 (PUG member): "...I am in the process of installing release 2 of Pascal 3.4 on our 6400 at Vogelback. I'm happy to see the improvements, especially the smaller core requirements. I've heard that Larry Liddiard has been trying to reduce the size....Pascal is not receiving the attention it deserves at Northwestern. I've added a Pascal mode to our interactive text editor as a first attempt in making Pascal use more attractive. We have several students interested in the language and one course this quarter

is using Wirth's Algorithms + Data Structures = Programs and running lab problems on our batch system...." (*11/22/76*)

Fleming M. Oliver, 213 Weddell #12, Sunnyvale, CA 94086 (PUG member): "Has anyone implemented Pascal on an Interdata 8/32? If so, who? I'm interested in the details." (*12/3/76*)

Ate Phung, Krefelder Str. 23, D-5100 Aachen, Germany (PUG member): "...As I am dealing with the implementation of Pascal to a Xerox Sigma 3, please send me any further informations about your experiences." (*12/8/76*)

Dean Schulz, INTEL Corporation, 3065 Bowers Avenue, Santa Clara, CA 95051 (PUG member): "...please inform me of all implementations of Pascal for microprocessors of which you are aware." (*11/29/76*)

Stephen C. Schwarm, E.I. du Pont de Nemours Co., 101 Beech St., Wilmington, DE 19898 (PUG member): "...we now have 150 persons on the DECUS (DEC Users Group) SIG Pascal mailing list! I'm preparing a newsletter which I will send shortly, and forward a copy to you as well....there is a big demand for an RSX-11 compiler. ...P4 is great for implementing! The problems with PDP-11s is that the compiler uses too much data space: 5K for the stack and 16K dynamic storage. On the other hand P-code translates well to PDP 11 code: 48K and this can be reduced to 32K with some hand optimization...." (*1/3/77*)

Manfred Seifert, Informatik III, Universitat Karlsruhe, Zirkel 2, D-7500 Karlsruhe, Germany (PUG member): "Our institute is running a PDP 11/45 and two PDP 11/34s linked together via DL-11E and DL-11B. We are using RSX-11M and RSX-11S, communication with DECNET-11 M/S. We are interested in a compiler under RSX-11M, or a compiler easily changeable to RSX support. We will use Pascal for programming IPC and decentralized control software in our local network." (*11/15/76*)

Rick Thomas, 408 Domer Ave., Takoma Park, MD 20012 (PUG member): "...We have a Univac 1108 installation at the University of Maryland and we just installed the Pascal/1100 compiler written at Naval Undersea Center by Mike Ball. We would be especially interested in an automatic Pascal source code indenter, for displaying the structure of someone else's code...." (*12/17/76*)

University of Washington, Seattle Computer Center Newsletter, August, 1976 shows that language processor statistics on its 6400/Cyber 73 system put Pascal at #2 behind Fortran (up from 5th the previous fiscal year). There was a decline in Simula usage over the same period. The Pascal increase was 65% from last year.

Tom Zwitter, Advanced Development Div., Building B, Ohio Nuclear Inc., 6000 Cochran Rd., Solon OH 44139 (PUG member): "...I spoke with Stephen Schwarm regarding their efforts and they hope to make an RSX-11 version available..." (*11/29*)

HERE AND THERE WITH PASCAL

(NEWS FROM MEMBERS, CONFERENCES, NEW BOOKS, APPLICATIONS PROGRAMS, ETC.)

HERE AND THERE WITH PASCAL

(NEWS FROM MEMBERS, CONFERENCES, NEW BOOKS, APPLICATIONS PROGRAMS, ETC.)

CONFERENCES

The Third Annual Computer Studies Symposium at the University of Southampton being organized by D.W. Barron and J.M. Mullins has added a speaker in the Application part of the program. Olivier Lecarme will speak on "Pascal and Portability". The symposium, entitled: Pascal, Implementation and Applications was fully described in Newsletter #6.

BOOKS AND ARTICLES

(* We really need someone to manage this section. *)

D.W. Barron reported in a letter dated 1 December, 1976 that there is a newly published book:

Introduction to PASCAL by C.A.G. Webster, Heyden, 1976. price: \$11.00, £5.50, DM35.00.

ROSTER 1/4/77 (NEW MEMBERS, CHANGED OR CORRECTED ADDRESSES AND PHONES)

THOMAS G. MCGINTY
DEPT. 330
FOXBORO CO.
38 NEPONSET AVE.
FOXBORO MA 02035
(617) 543-8750 X2031

JOHN CASEY
DEPARTMENT OF MATHEMATICS
NORTHEASTERN UNIVERSITY
360 HUNTINGTON AVENUE
BOSTON MA 02115
(617) 437-2450

ROBERT E. WELLS
BOLT BERANEK AND NEWMAN INC.
50 MOULTON STREET
CAMBRIDGE MA 02138
(617) 491-1850

ROY A. WILSKER
27 BENEFIT STREET
WALTHAM MA 02154
(617) 899-6638

FRED EILENSTEIN
68 SPRING STREET
WATERTOWN MA 02172
(617) 924-2248

G. M. SHANNON
LINCOLN LAB
J-146
M.I.T.
244 WOOD STREET
LEXINGTON MA 02173
(617) 862-5500 X5719

KEN POLAKOWSKI
50 VILLAGE GREEN
BUDD LAKE NJ 07828
(201) 347-4375

RICHARD B KIEBURTZ
DEPT. OF COMPUTER SCI.
SUNY AT STONY BROOK
STONY BROOK NY 11794
(516) 246-5987

S. KAMAL ABDALI 12181
GARY S. ANDERSON 98043
ATTN: A.D.R. 38000 FRANCE
ATTN: CENTRAL LIBRARY 3001 AUSTRALIA
ATTN: COMPUTER CENTER USER SERVICES 98105
ATTN: COMPUTER UNIT CV4 7AL UNITED KINGDOM
ATTN: THE LIBRARY ISRAEL
HENRY R. BAUER III 82071
HERMAN BERG 53703
SCOTT BERTILSON 55455
JEAN BEZIVIN 35031 FRANCE
I. N. BLAVINS 5006 AUSTRALIA
C. BRON THE NETHERLANDS
J. A. CAMPBELL 2308 AUSTRALIA
JOHN CASEY 02115
JOHN E. COLLINS 55101
RICHARD CORE 94088
B. J. CORNELIUS HU6 7RX UNITED KINGDOM
LINDA E. CROLEY 94304
DENNIS DANCE 72204
DAVID DEMOREST 98124
PIERRE DESJARDINS 101 CANADA
FRED EILENSTEIN 02172
JACQUES FARRE 75230 FRANCE
EDWARD E FERGUSON 76201
TED FISHMAN 75222
DAVID C. FITZGERALD 91740
J. J. FLORENTIN WC1E 7HX UNITED KINGDOM
JIM FONTANA 92704
DAVID N. GRAY 78769
HILMAR GUTFELDT CH-3000 SWITZERLAND
C. C. HANDLEY 4000 SOUTH AFRICA
KAY A. HANSBOROUGH 87544
KAY HARRISON N2L 3G1 CANADA
KEVIN HAUSMANN 55113
S. T. HEIDELBERG 94550
RICHARD HENDRICKSON 55420
TERUO HIKITA 113 JAPAN
H.-J. HOFFMANN D-6100 GERMANY
TIMOTHY J. HOFFMANN 55455
WILLIAM C. HOPKINS 19174
PEI HSIA 35807
R. I. JOHNSON 58202
ED KATZ 70504
RICHARD B KIEBURTZ 11794
JOHN A. LAHRETT 2308 AUSTRALIA
ROBERT M. LANSFORD 91107
ROBERT A. LAWLER 55165
MIKE LEMON 61738
GEORGE LIGLER 75080
GARY LINDSTROM 84112
BRUCE LINK 87115
LEON LUKASZEWICZ 00901 POLAND
ORLANDO S. MADRIGAL 95926
LARS MAGNUSSON S-751 21 SWEDEN
CRAIG MAUDLIN 92037
KONRAD MAYER A-1150 AUSTRIA
THOMAS G. MCGINTY 02035
BRIAN MEEK W8 7AH UNITED KINGDOM
B. T. MITCHELL KY16 UNITED KINGDOM

CHARLES G. MOORE 48106
LARS G. MOSSBERG S-461 01 SWEDEN
BERNHARD NEBEL 2 GERMANY
BERGT NORDSTROM S-402 20 SWEDEN
JOHN L. NORSTAD 60201
FLEMING M. OLIVER 94086
PAUL O-BRIEN 97210
TRUMAN C. PEWITT 60439
ATE PHUNG D-5100 GERMANY
KEN POLAKOWSKI 07828
FRED W. POWELL 24401
GEORGE H. RICHMOND 80309
CLAES RICEBEY S-161 54 SWEDEN
N ROBINSON HA6 30Z UNITED KINGDOM
DON H. ROWLAND 87109
MARK RUSTAD 55112
WILLIAM A. RUTISER 20052
HORST SANTO D-5205 GERMANY
ROSS D. SCHMIDT 55343
DEAN SCHULZ 95051
MANFRED SEIFERT D-7500 GERMANY
G. M. SHANNON 02173
DAVID ELLIOT SHAW 94301
JOHN M. SHAW 20014
J. B. SLATER NW3 7ST UNITED KINGDOM
GORDON STUART V8P 5JZ CANADA
MENACHEM SZUS ISRAEL
RAMON TAN 18104
DIDIER THIBAUT 75005 FRANCE
RICK THOMAS 20012
RON THOMAS 55425
SEVED TORSTENDAHL S-145 72 SWEDEN
EDWIN TSE H9R 1G1 CANADA
W. TYLER 95060
INDULIS VALTERS 55455
STEVEN W. WEINGART 55113
ROBERT E. WELLS 02138
ROY A. WILSKER 02154
DENIS M. WILSON AB9 2UB UNITED KINGDOM
GREGORY J. WINTERHALTER 48103
JOHN M. WOBUS 13210
ANDREW HARRIS ZIMMERMAN 94086
TOM ZWITTER 44139

S. KAMAL ABDALI
DEPT. OF MATHEMATICAL SCIENCES
RENSSELAER POLYTECHNIC INSTITUTE
TROY NY 12181
(518) 270-6558

JOHN M. WOBUS
453 WESTCOTT ST. APT. 1
SYRACUSE NY 13210
(315) 472-4923

RAMON TAN
2345 UNION ST.
ALLEN TOWN PA 18104
(215) 434-5432

WILLIAM C. HOPKINS
DEPT. OF COMP. AND INFO. SCI.
U OF PENNSYLVANIA
PHILADELPHIA PA 19174
(215) 243-8549

RICK THOMAS
408 DOMER AVENUE
TAKOMA PARK MD 20012
(301) 565-2678

JOHN M. SHAW
BLDG 36 / ROOM 2A29
NATIONAL INSTITUTES OF HEALTH
BETHESDA MD 20014
(301) 496-3204

WILLIAM A. RUTISER
CAAC
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON DC 20052
(202) 676-6525

FRED W. POWELL
INNOVATIVE MANAGEMENT SYSTEMS
865 MIDDLEBROOK AVENUE
STAUNTON VA 24401

PEI HSIA
COMPUTER SCIENCE PROGRAM
U OF ALABAMA AT HUNTSVILLE
P.O. BOX 1247
HUNTSVILLE AL 35807

TOM ZWITTER
ADVANCED DEVELOPMENT DIV.
BUILDING B
OHIO NUCLEAR INC.
6000 COCHRAN RD.
SOLON OH 44139

GREGORY J. WINTERHALTER
5148 DEXTER RD.
ANN ARBOR MI 48103

CHARLES G. MOORE
NETWORK SERVICES INC.
175 JACKSON PLAZA
ANN ARBOR MI 48106
(313) 426-2620

HERMAN BERG
108 E. DAYTON
MADISON WI 53703
(608) 251-1910

JOHN E. COLLINS
BLDG 235 F247
3M CENTER
ST. PAUL MN 55101
(612) 736-0778

MARK RUSTAD
585 HARRIET AVE #213
ST. PAUL MN 55112

KEVIN HAUSMANN
MECC
2520 W. BROADWAY
LAUDERDALE MN 55113
(612) 376-1119

STEVEN W. WEINGART
MS 4753
SPERRY-UNIVAC
2276 HIGHCREST DRIVE
ROSEVILLE MN 55113
(612) 633-6170 X3748

ROBERT A. LAWLER
MS U2423
UNIVAC PARK
P.O. BOX 3525
ST. PAUL MN 55165
(612) 456-3107

ROSS D. SCHMIDT
MN 11-2120
HONEYWELL INC.
600 2ND ST. NO.E.
HOPKINS MN 55343
(612) 542-6741

RICHARD HENDRICKSON
CRAY RESEARCH INC.
7850 METRO PARKWAY SUITE 213
MINNEAPOLIS MN 55420
(612) 854-7472

RON THOMAS
DATA 100 CORPORATION
7725 WASHINGTON AVE. S.
MINNEAPOLIS MN 55425
(612) 941-6500

SCOTT BERTILSON
UNIVERSITY COMPUTER CENTER
227 EXP. ENGR.
U OF MINNESOTA
MINNEAPOLIS MN 55455
(612) 376-5262 (WORK)
(612) 729-0059 (HOME)

TIMOTHY J. HOFFMANN
UNIVERSITY COMPUTER CENTER
227 EXP. ENGR.
U OF MINNESOTA
MINNEAPOLIS MN 55455
(612) 373-6957 (HOME)
(612) 376-5262 (WORK)

INDULIS VALTERS
MISRC
93 BLEGEN HALL
U OF MINNESOTA
WEST BANK
MINNEAPOLIS MN 55455
(612) 341-4430 (HOME)

R. I. JOHNSON
COMP. SCI. DEPT.
U OF NORTH DAKOTA
BOX 3181 UNIVERSITY STATION
GRAND FORKS ND 58202
(701) 777-4107

JOHN L. NORSTAD
VOGELBACK COMPUTING CENTER
NORTHWESTERN UNIVERSITY
2129 SHERIDAN RD.
EVANSTON IL 60201
(312) 492-5369

TRUMAN C. PEWITT
APPLIED MATH DIVISION
BLOG. 221
ARGONNE NATIONAL LABORATORY
9700 SOUTH CASS AVENUE
ARGONNE IL 60439
(312) 739-7711

MIKE LEMON
168 WEST THIRD STREET
EL PASO IL 61738
(309) 527-4342

ED KATZ
COMPUTER SCIENCE DEPT.
U OF SOUTHWESTERN LOUISIANA
BOX 4-4330 USL STATION
LAFAYETTE LA 70504
(318) 233-6840
(318) 233-6767

DENNIS DANCE
COMPUTER SCIENCE DEPT.
UNIVERSITY OF ARKANSAS AT LITTLE ROCK
33RD AND UNIVERSITY
LITTLE ROCK AR 72204
(501) 569-3252

GEORGE LIGLER
1000 W. SPRING VALLEY RD. APT. 263
RICHARDSON TX 75080
(214) 231-0825

TED FISHMAN
TEXAS INSTRUMENTS
P.O. BOX 6015 (MS 295)
DALLAS TX 75222
(214) 349-3028

EDWARD E FERGUSON
1222 AUSTIN AVE
DENTON TX 76201
(214) 238-4092

DAVID N. GRAY
MS 2188
TEXAS INSTRUMENTS
P.O. BOX 2909
AUSTIN TX 78769
(512) 258-5121

GEORGE H. RICHMOND
COMPUTING CENTER
UNIVERSITY OF COLORADO
3645 MARINE STREET
BOULDER CO 80309
(303) 492-8131

HENRY R. BAUER III
COMPUTER SCIENCE DEPT.
UNIVERSITY OF WYOMING
BOX 3682
LARAMIE WY 82071
(307) 766-5134

GARY LINDSTROM
COMPUTER SCIENCE DEPT.
U OF UTAH
SALT LAKE CITY UT 84112
(801) 581-8224

DON H. ROWLAND
5805 TORREON DR.
ALBUQUERQUE NM 87109
(505) 821-9207 (HOME)
(505) 264-9149 (OFFICE)

BRUCE LINK
DIVISION 1712
SANDIA LABORATORIES
ALBUQUERQUE NM 87115

KAY A. HANSBOROUGH
2377B 45TH ST.
LOS ALAMOS NM 87544
(505) 662-9369 (HOME)
(505) 667-6034 (OFFICE)

ROBERT M. LANSFORD
BURROUGHS CORPORATION
3620 GREENHILL ROAD
PASADENA CA 91107
(213) 351-0206

DAVID C. FITZGERALD
652 S. CULLEN
GLENORA CA 91740
(213) 335-6055

CRAIG MAUDLIN
3161 -F- VIA ALICANTE DR.
LA JOLLA CA 92037
(714) 452-8716

JIM FONTANA
CONTROL DATA CORPORATION
3519 W. WARNER AVE.
SANTA ANA CA 92704
(714) 754-4102

FLEMING M. OLIVER
213 WEDDELL APT. 12
SUNNYVALE CA 94086

ANDREW HARRIS ZIMMERMAN
550 NORTH FAIR OAKS AVE. APT. 14
SUNNYVALE CA 94086

RICHARD CORE
PO BOX 61628
SUNNYVALE CA 94088
(408) 735-8400 X233

DAVID ELLIOT SHAW
STRUCTURED SYSTEMS CORP.
427 EMBARCADERO ROAD
PALO ALTO CA 94301
(415) 321-8111

LINDA E. CROLEY
BNR INC.
3174 PORTER DR.
PALO ALTO CA 94304
(415) 494-3942 X40 OR 61

S. T. HEIDELBERG
DIVISION 8323
SANDIA LABORATORIES
LIVERMORE CA 94550
(415) 455-2179

DEAN SCHULZ
INTEL CORPORATION
3065 BOWERS AVENUE
SANTA CLARA CA 95051
(408) 246-7501

W. TYLER
200 SEABORG PLACE
SANTA CRUZ CA 95060
(408) 925-0206

ORLANDO S. MADRIGAL
DEPARTMENT OF COMPUTER SCIENCE
CALIFORNIA STATE UNIVERSITY AT CHICO
CHICO CA 95926
(916) 895-6442

PAUL O-BRIEN
P.O. BOX 10572
PORTLAND OR 97210
(503) 244-7538

GARY S. ANDERSON
JOHN FLUKE MFG. CO. INC.
P.O. BOX 43210 M.S. 16
MOUNTLAKE TER WA 98043
(206) 774-2211 X353

ATTN: COMPUTER CENTER USER SERVICES
UNIVERSITY OF WASHINGTON
3737 BROOKLYN AVE. N.E. RM 15
SEATTLE WA 98105

DAVID DEMOREST
M/S 8M-71
BOEING COMPUTER SERVICES
P.O. BOX 24346
SEATTLE WA 98124
(206) 244-6923
(206) 773-2019

J. A. CAMPBELL
MATHEMATICS DEPT.
NEWCASTLE UNIVERSITY
N.S.W. 2308
AUSTRALIA

JOHN A. LAMBERT
COMPUTING CENTRE
UNIVERSITY OF NEWCASTLE
N.S.W. 2308
AUSTRALIA

ATTN: CENTRAL LIBRARY
FLOOR 1 CASEY WING
ROYAL MELBOURNE INSTITUTE OF TECHNOLOG
376-392 SWANSTON STREET
MELBOURNE VICTORIA 3001
AUSTRALIA

I. N. BLAVINS
KATHLEEN LUMLEY COLLEGE
FINNIS STREET
NORTH ADELAID S.A. 5006
AUSTRALIA

KONRAD MAYER
REICHSAPFELG 13/8
VIENNA A-1150
AUSTRIA

EDWIN TSE
525 DELMAR ST.
POINTE CLAIRE QUEBEC H9R 1G1
CANADA
(514) 697-1320

KAY HARRISON
COMPUTER CENTER
1088B M AND C
U OF WATERLOO
WATERLOO ONTARIO N2L 3G1
CANADA

GORDON STUART
TECHNICAL AND VOCATIONAL INST.
CAMOSUN COLLEGE
1950 LANSDOWNE RD.
VICTORIA B.C. V8P 5J2
CANADA
(604) 592-1281 X248

PIERRE DESJARDINS
INFORMATIQUE
UNIVERSITE DE MONTREAL
C.P. 6128
MONTREAL QUEBEC 101
CANADA
(514) 343-6463

JEAN BEZIVIN
DEPARTEMENT DE MATHÉMATIQUES & INFORMATIQUE
UNIVERSITE DE RENNES
RENNES CEDEX 35031
FRANCE
36.48.15

ATTN: A.D.R.
CHAMBRE DE COMMERCE ET D-INDUSTRIE
6 BLD GAMBETTA
GRENOBLE 38000
FRANCE

DIDIER THIBAUT
17 RUE GAY-LUSSAC
PARIS 75005
FRANCE
527 16 85

JACQUES FARRE
T 55.65
INSTITUT DE PROGRAMMATION
4 PLACE JUSSIEU
PARIS CEDEX 05 75230
FRANCE
336 25 25 X58 77

ATE PHUNG
KREFELDER STR. 23
AACHEN D-5100
GERMANY

HORST SANTO
GMD-IPES
POSTFACH 1240
ST.AUGUSTIN 1 D-5205
GERMANY

H.-J. HOFFMANN
FACHBEREICH INFORMATIK
TECHNISCHE HOCHSCHULE
STUEBENPLATZ 12
DARMSTADT D-6100
GERMANY

MANFRED SEIFERT
INFORMATIK III
UNIVERSITÄT KARLSRUHE
ZIRKEL 2
KARLSRUHE D-7500
GERMANY
0721/608-3982

BERNHARD NEBEL
STEGLITZER STR. 17F
HAMBURG 70 2
GERMANY
040/664911

ATTN: THE LIBRARY
MINISTRY OF DEFENCE
P.O.BOX 962
HAIFA
ISRAEL

MENACHEM SZUS
ART AND SCIENCE
BEZALEL ACADEMY OF ART AND DESIGN
10 SHMUEL HANAGID ST.
JERUSALEM
ISRAEL

TERUO HIKITA
DEPT. OF INFO. SCI.
U OF TOKYO
TOKYO 113
JAPAN
03-812-2111 X2947

LEON LUKASZEWICZ
COMPUTATION CENTRE
POLISH ACADEMY OF SCIENCE
WARSZAWA PKIN 00901
POLAND
200211 X2225

C. C. HANDLEY
DEPT. OF COMPUTER SCIENCE
UNIVERSITY OF DURBAN-WESTVILLE
P/BAG X54001
DURBAN 4000
SOUTH AFRICA
821211 X138

SEVED TORSTENDAHL
TOMTBORGAV 279
NORSBORG S-145 72
SWEDEN

CLAES RICKEY
HEDEBYVAGEN 5
URUMMA S-161 54
SWEDEN
08/37 65 37

BERGT NORDSTROM
DEPARTMENT OF COMPUTER SCIENCES
CHALMERS INSTITUTE OF TECHNOLOGY
GOTEBERG S-402 20
SWEDEN

LARS G. MOSSBERG
VOLVO FLYGMOTOR AB
BOX 136
TROLLHATTEN S-461 01
SWEDEN

LARS MAGNUSSON
INSTITUTE OF TECHNOLOGY
UPPSALA UNIVERSITY
BOX 534
UPPSALA S-751 21
SWEDEN
018-10 04 70

HILMAR GUTFELDT
RESEARCH AND DEVELOPMENT-DEPT. 82
HASLER LTD.
BELPSTRASSE 23
BERNE 14 CH-3000
SWITZERLAND
031 65 21 11

C. BRON
DEPT. OF ELECTRICAL ENGINEERING
TECHNISCHE HOOGESCHOOL TWENTE
POSTBUS 217
ENSCHDE
THE NETHERLANDS
(031) 53 894451

DENIS M. WILSON
DEPARTMENT OF COMPUTING SCIENCE
UNIVERSITY OF ABERDEEN
KING-S COLLEGE
OLD ABERDEEN SCOTLAND AB9 2UB
UNITED KINGDOM

ATTN: COMPUTER UNIT
COMPUTER CENTER
UNIVERSITY OF WARWICK
COVENTRY ENGLAND CV4 7AL
UNITED KINGDOM
(0203) 24011 X2754

N ROBINSON
1 THE FAIRWAY
NORTHWOOD MIDDLESEX HA6 3DZ
LONDON ENGLAND
UNITED KINGDOM

B. J. CORNELIUS
DEPT. OF COMP. STUDIES
UNIVERSITY OF HULL
HULL ENGLAND HU6 7RX
UNITED KINGDOM

B. T. MITCHELL
COMPUTING LABORATORY
UNIVERSITY OF ST. ANDREWS
NORTH HAUGH ST. ANDREWS
FIFE SCOTLAND KY16
UNITED KINGDOM

J. B. SLATER
COMPUTER UNIT
WESTFIELD COLLEGE
KIDDERPORE AVENUE
LONDON ENGLAND NW3 7ST
UNITED KINGDOM

J. J. FLORENTIN
DEPARTMENT OF COMPUTER SCIENCE
BIRKBECK COLLEGE
MALET STREET
LONDON ENGLAND WC1E 7HX
UNITED KINGDOM

BRIAN MEEK
COMPUTER UNIT
QUEEN ELIZABETH COLLEGE
CAMPDEN HILL ROAD
LONDON ENGLAND W8 7AH
UNITED KINGDOM

ARTICLES

(FORMAL SUBMITTED CONTRIBUTIONS)

LIFE, LIBERTY AND THE PURSUIT OF UNFORMATTED INPUT

D.W. BARRON AND J.M. MULLINS

University of Southampton

IN PUGN^W5, Eisenberg presents three examples which, he claims, demonstrate the necessity for formatted input. This note attempts to demolish those claims.

Example 1 is concerned with survey analysis. 75 candidates are rated on a scale 1 to 5, and the observations from each individual taking part in the survey are punched as a contiguous stream of 75 digits. Formatted input allows this to be read as 75 integers: extracting the integer values by reading characters and using `ord(ch) - ord('0')` is said to "detract from the exercise ... the student is not interested in the use of "ord" or strings...."

The student needs to realise that dealing with this sort of data is messy, so that in future he can tell people designing surveys to prepare their data in a more palatable manner, instead of producing unreadable and inherently error prone sequences of digits. (Anyone designing a survey ought to consult the person responsible for analysing the results before the data is punched. They can then be told that redundancy in the form of separators allows error checking.)

Example 2 "Your Ph.D. Advisor" has produced a card deck in which "columns 11 to 70 contain 20 3-digit numbers...".

See the remarks on example 1 above. If your Ph.D. Advisor really produces data like this, he ought not to be allowed to advise Computer Science Ph.D. students. We are asked to note that omitting blanks "saves keypunching time and storage space, both of which are the equivalent of money". Balderdash. Do the sums, and compare the cost with the time of the (expensive) programmer sorting out the resultant mess. (Incidentally, anyone who tells us to keypunch real numbers in a contiguous stream to avoid "wasting time typing spaces" will be told, politely but firmly, to go get his head examined.)

Example 3 is presented as a situation where formatted input "is a near necessity". The problem is to extract the three fields from the 7-digit code on a pair of jeans, where the first three digits code the style, the next two code the color, and the last two digits code the material. Who needs formatted input? How about:

```
var code, style, color, material : integer;
read(code);
material := code mod 100;
color    := (code div 100) mod 100;
style    := code div 10000;
```

This method works as long as we can read the largest code, 7070436, which means we need at least 24-bit arithmetic. Doing it this way in a class has the advantage that the student will understand mod and div at the end of it, as well as learning something about positional coding systems, the representation of integers, and the significance of maxint.

If we want to avoid the maxint restriction, we have to use the dreaded 'ord'. But is it so terrible? If we have defined type `digit = 0..9` and we are prepared to take the accuracy of the input on trust (we shouldn't, but formatted input schemes do), all we need is

```
function nextdigit : digit;
var ch : char;
begin
read (ch);
nextdigit := ord(ch) - ord('0')
end;
```

If the student is dealing with these kind of codes it will do him no harm to use `ord`, since in any case he ought to appreciate the difference between the character '3' and the number 3.

ARTICLES

(FORMAL SUBMITTED CONTRIBUTIONS)

Finally, if we want to be fancy we can define

```
type field = (onecolumn, twocolumns, threecolumns);
and function formattedread (f : field): integer;
begin
  case f of
    onecolumn : formattedread := nextdigit;
    twocolumns : formattedread := 10* nextdigit + nextdigit;
    threecolumns : formattedread := 100* nextdigit +
      10* nextdigit + nextdigit
  end;
end;
```

OK, this only does what formatted input would have done. But we can amend nextdigit to catch input errors, and educationally it is far better to learn to do it this way than it is to learn how to construct format statements.

There is only one moral to be drawn from the defence of formatted input:

WARNING: FORTRAN CAN IMPAIR YOUR JUDGEMENT

(*Received 11/29/76*)

PRNTPLT - PASCAL PRINTER PLOTTER

(* a set of subprograms for producing two-dimensional *)
(* (X,Y) plots on ordinary printers *)

(* by Herb Rubenstein, Research Assistant

User University Computer Center
Callable University of Minnesota *)
Procedures: SCLINITIAL
SCLWINDOW
SCLPLOT
SCLPRINT

Language: PASCAL 6000.

Computer System: KRONOS 2.1 CDC CYBER 74.

Location: University of Minnesota.

Programmer: Herbert Rubenstein.

Documentor: same.

Date: September, 1976. (revised - December, 1976)

Contents: A.) General Description.
B.) Using the PRINTER PLOTTER.
C.) Programming Examples.
D.) User Hints.
E.) Notes.

A. General Description:

This PRINTER PLOTTER consists of four PASCAL callable procedures: SCLINITIAL, SCLWINDOW, SCLPLOT, and SCLPRINT. A high speed line-printer, 1004 remote job entry terminal, or a 132 column hard-copy interactive terminal (such as a DECwriter) is used for output. The plots are X-Y graphs scaled to fit a single sheet of printer-paper. X values run horizontally while Y values are vertical. Axis labels and axes are automatically set up. The user need only provide data points, titles, and plotting characters. Nearly any range of data is suitable. Plots overlaid on one another (multiplots) can be generated just as easily as single plots. Using SCLWINDOW, (expansions) can be made. That is, a tiny piece of a plot can literally be blown-up and made into a new plot. The PASCAL PRINTER PLOTTER was created primarily because the printer-plotting routines, available in the FORTRAN library do not offer the features and simplicity described herein. Besides, they are not compatible with PASCAL anyway.

B. Using the PRINTER PLOTTER:

The PASCAL PRINTER PLOTTER is easy to use. Only a few declarations are necessary. Four separate procedures must be considered: SCLINITIAL, SCLWINDOW, SCLPLOT, and SCLPRINT.

Single plots are made by calling the basic procedures once, first SCLINITIAL, then SCLPLOT, and finally SCLPRINT. Multiplots are made by repetitively calling SCLPLOT. The first call establishes the scale, and all further calls will then be handled according to that scale. Any points lying outside the boundaries established by either SCLPLOT (first call), or a call to SCLWINDOW will be ignored.

SCLWINDOW acts as if it were the first call to SCLPLOT except there is no plotting. SCLWINDOW provides the expansion facility by allowing the user to specify ranges of X and Y values directly.

B.1. Variable and Type Declarations:

Certain declarations are required by the PASCAL PRINTER PLOTTER. The user should insert them into his MAIN PROGRAM. The most important is an ARRAY of 700 words which SCLPLOT uses as a large work area. Since the four procedures also communicate through this ARRAY, it is very important that it not be altered by the user. Identifiers used in the following illustration are only suggested. In other words, the large work area, which will be referred to as IMAGE, could just as well be called XYZ.

Example B.1

```
PROGRAM TESTPLOTTER(INPUT,OUTPUT);
CONST;
  NUM=50; (*MAX NUMBER OF DATA POINTS TO BE PLOTTED*)
TYPE
  SCRATCH = ARRAY [1..700] OF INTEGER;
  ARDATA = ARRAY [1..NUM] OF REAL;
  CH60 = PACKED ARRAY [1..60] OF CHAR;
VAR
  X : ARDATA; (*ARRAY OF X-VALUES*)
  Y : ARDATA; (*ARRAY OF CORRESPONDING Y-VALUES*)
  N : INTEGER; (*NUMBER OF DATA POINTS TO BE PLOTTED*)
  IMAGE : SCRATCH; (*WORK AREA FOR PASCAL PRINTER PLOTTER*)
  TITLE : CH60; (*TITLE OF THE PLOT *)
  XLEG : CH60; (*HORIZONTAL HEADING FOR THE X-AXIS*)
  YLEG : CH60; (*VERTICAL HEADING FOR THE Y-AXIS*)
  CH : CHAR; (*PLOTTING CHARACTER*)
```

Note:

It is important that SCRATCH, ARDATA, and CH60, or any other names used for these, be defined exactly as shown above. SCLINITIAL, SCLWINDOW, SCLPLOT, and SCLPRINT expect this and cannot function otherwise. Also, the variables: TITLE, XLEG, YLEG, CH, and N are optional. This depends upon the particular form of PROCEDURE usage. See section B.3 and example B.3.3.

B.2. Procedure Declarations:

The user should also declare the four procedures: SCLINITIAL, SCLWINDOW, SCLPLOT, and SCLPRINT in the MAIN PROGRAM. Notice that IMAGE, PROFILE, X, and Y are the only variable parameters. X1, X2, Y1, Y2, N, CH, TITLE, XLEG, and YLEG are value parameters. Value parameters can be specified as constants, directly in a PROCEDURE call. This can help eliminate extra variables and also can be an aid to readability.

PROCEDURE declarations must appear as shown below. Again, identifiers can be anything convenient, however positioning is extremely important. For example, in SCLPLOT, the parameters: IMAGE, N, X, Y, CH must appear in that order. X, Y, N, CH, IMAGE is syntactically incorrect.

Example B.2

```
PROCEDURE SCLINITIAL(VAR IMAGE:SCRATCH);EXTERN;
PROCEDURE SCLWINDOW(VAR IMAGE:SCRATCH;
  X1,X2,Y1,Y2:REAL);EXTERN;
PROCEDURE SCLPLOT(VAR IMAGE:SCRATCH;
  N:INTEGER;
  VAR X,Y:ARDATA;
  CH:CHAR);EXTERN;
PROCEDURE SCLPRINT(VAR IMAGE:SCRATCH;VAR PROFILE:TEXT;
  TITLE,XLEG,YLEG:CH60);EXTERN;
```

Note:

PROCEDURE declarations must be made under the default compiler X option, that is, X4 for PASCAL 6000.

B.3. Typical PROCEDURE Usage:

The following are four examples of typical PROCEDURE usage. The first three are single plots. The fourth is a multiplot. The second is an expansion of the first using SCLWINDOW. The third shows how convenient it can be to substitute constants directly as ACTUAL PARAMETERS, thus eliminating extra variables. Notice that TITLE, XLEG, and YLEG are actually strings. Strings are defined in PASCAL as PACKED ARRAYS OF CHAR. Also, note that SCLPRINT writes the graph on the file, OUTPUT. A different external file, other than OUTPUT, can be used if desired.

Example B.3.1

```
-----
SCLINITIAL(IMAGE);          (* INITIALIZE *)
SCLPLOT(IMAGE,N,X,Y,CH);    (* PLOT *)
SCLPRINT(IMAGE,OUTPUT,TITLE,XLEG,YLEG); (* PRINT *)
```

Example B.3.2

```
-----
SCLINITIAL(IMAGE);          (* INITIALIZE *)
SCLWINDOW(IMAGE,X1,X2,Y1,Y2); (* EXPAND *)
SCLPLOT(IMAGE,N,X,Y,CH);    (* PLOT *)
SCLPRINT(IMAGE,OUTPUT,TITLE,XLEG,YLEG); (* PRINT *)
```

Example B.3.3

```
-----
SCLINITIAL(IMAGE);
SCLPLOT(IMAGE,47,X,Y,"P"); (*CONSTANTS ARE SUBSTITUTED FOR N AND CH*)
SCLPRINT(IMAGE,OUTPUT,
".....THIS IS THE TITLE.....",
".....THIS IS THE X-LEGEND.....",
".....THIS IS THE Y-LEGEND.....");
(*CONSTANTS ARE SUBSTITUTED FOR TITLE, XLEG, AND YLEG*)
```

Example B.3.4

```
-----
SCLINITIAL(IMAGE);
SCLPLOT(IMAGE,N,X,Y,"A"); (* ESTABLISH SCALING *)
SCLPLOT(IMAGE,N,X,Y,"B"); (*ALL FURTHER REFERENCES TO SCLPLOT
ARE DEPENDENT UPON THE SCALE JUST
ESTABLISHED. *)
.
.
.
.
.
SCLPLOT(IMAGE,N,X,Y,"Y");
SCLPLOT(IMAGE,N,X,Y,"Z");
SCLPRINT(IMAGE,OUTPUT,TITLE,XLEG,YLEG); (*PRINTS THE MULTIPLY*)
```

C. Programming Examples:

Three different programming examples will be given at the end of this writeup. The first is an example of a single plot. It shows what happens if the data is too large or small to have properly displayed axis labels. A scale factor message is printed beside each axis affected. The second is a multiplot. Finally, the third is an expansion of the second showing where B, C, and D intersect. The range of X-values is from 4.5 to 5.5 while the range of Y-values is from 0.4 to 0.6.

Notice that when using SCLWINDOW for expansion purposes, the more data points provided, the more resolution possible. That is, if the user generates more X and Y values, (by making delta X smaller), more expansions are possible. Expansions of expansions are easily obtained in this manner. If the user is running out of data points after repetitively expanding a plot, then decreasing delta X and generating more points will help.

In examples C.2 and C.3 be sure to notice how Y is declared as an ARRAY of ARRAYS of Y-values. This makes overplotting convenient in a looping situation. Do not forget that it is either the first call to SCLPLOT, or a single call to SCLWINDOW which determines the overall scale. Any calls thereafter are treated according to that scale.

Do not call SCLWINDOW more than once. Nothing will happen except time will be wasted. For the same reason, do not call SCLWINDOW if SCLPLOT has previously been called. Also, if either the X-values or the Y-values for SCLWINDOW are accidentally given as being equal, no scaling will occur. The first SCLPLOT call would then provide the scaling. The parameters following IMAGE can be in any numerical order as long as the first two are X-values and the second two are Y-values.

D. Hints, Cautions, Errors:

There is one mistake a user could make which can cause unpredictable results, that is, failure to call SCLINITIAL.

The usual syntax precautions apply here as well as in any PASCAL program. Make sure that the parameters are of the right TYPE and see that they are properly positioned. Do not forget that the variable parameters are: IMAGE, PROFILE, X, and Y, and that the value parameters are: X1, X2, Y1, Y2, N, CH, TITLE, XLEG, and YLEG. Also, if the user is manipulating compiler options, the X option must be set to X4 for PROCEDURE declarations.

