HP 2000 SERIES
CONTRIBUTED LIBRARY

TIME-SHARED BASIC/2000
PROGRAM DOCUMENTATION

VOLUME I

(100) DATA HANDLING
(200) TESTING, DEBUGGING AND PROGRAMMING AIDS

36000-91001

Printed in U.S.A. June 1974

HEWLETT PACKARD

TIME-SHARED BASIC/2000 CONTRIBUTED LIBRARY HANDBOOK

VOLUME I

(100) DATA HANDLING
(200) TESTING, DEBUGGING AND PROGRAMMING AIDS

HEWLETT-PACKARD, SOFTWARE CENTER, 11000 Wolfe Road, Cupertino, California 95014 Area Code 408 257-7000 TWX 910-338-0221

36000-91001 6/74 Volume I

CLASSIFICATION CODE CATEGORY

(Not all categories have programs. Please refer to the INDEX to HP BASIC Program Library for available programs in HP BASIC)

100 DATA HANDLING (VOLUME I)

101 EDITING
102 INFORMATION STORAGE AND RETRIEVAL
103 TABLE HANDLING
104 CHARACTER/SYMBOL MANIPULATION
105 CODE/RADIX CONVERSION
106 DUPLICATION
107 SORTING AND MERGING
108 DATA HANDLING UTILITIES
109 MEDIA CONVERSION
110 FILE MANAGEMENT
112 SPECIAL FORMAT DATA TRANSFER
114 PLOT ROUTINES IN HP BASIC

200 TESTING, DEBUGGING AND PROGRAMMING AIDS (VOLUME I)

201 TRACING
202 INSTRUMENT TEST
203 DISC/DRUM EQUIPMENT TEST
204 MAGNETIC TAPE EQUIPMENT TEST
205 GRAPHIC EQUIPMENT TEST
206 MEMORY SEARCH AND DISPLAY
207 DUMPING
208 CORE STORAGE TEST
209 CENTRAL PROCESSING UNIT TEST
210 BREAK POINTS
211 DEBUGGING AIDS
212 PROGRAMMING AIDS
213 PAPER TAPE EQUIPMENT TEST
214 PUNCH CARD EQUIPMENT TEST
215 PRINTER EQUIPMENT TEST
216 A/D - D/A EQUIPMENT TEST
217 TELECOMMUNICATIONS EQUIPMENT TEST
218 SPECIAL DEVICE EQUIPMENT TEST
219 DATA ACQUISITION SYSTEMS TEST

300 MATH AND NUMERICAL ANALYSIS (VOLUME II)

301 MATHEMATICS, GENERAL
302 EXTENDED-PRECISION ARITHMETIC
303 COMPLEX ARITHMETIC
304 BCD/ASCII ARITHMETIC
305 BOOLEAN ALGEBRA
306 FUNCTIONS, COMPUTATION OF
307 INTERPOLATION/EXTRAPOLATION
309 CURVE FITTING
310 NUMERICAL INTEGRATION
311 POLYNOMIALS AND POLYNOMIAL EQUATIONS
312 MATRIX OPERATIONS
313 EIGENVALUES AND EIGENVECTORS
314 SYSTEMS OF LINEAR EQUATIONS
315 SYSTEMS OF NON-LINEAR EQUATIONS
316 INTEGRAL TRANSFORMS
317 NUMERICAL DIFFERENTIATION
318 ORDINARY DIFFERENTIAL EQUATIONS
319 PARTIAL DIFFERENTIAL EQUATIONS

400 PROBABILITY AND STATISTICS (VOLUME II)

401 UNIVARIATE AND MULTIVARIATE PARAMETRIC STATISTICS
402 TIME SERIES ANALYSIS
403 DISCRIMINANT ANALYSIS
404 REGRESSION ANALYSIS
405 RANDOM NUMBER GENERATORS
406 PROBABILITY DISTRIBUTION SAMPLING
407 NON-PARAMETRIC STATISTICS
408 STATISTICS, GENERAL
409 CORRELATION ANALYSIS
410 ANALYSIS OF VARIANCE AND COVARIANCE
411 FACTOR ANALYSIS
412 SCALING
413 GENERAL PROBABILITY

500 SCIENTIFIC AND ENGINEERING APPLICATIONS (VOLUME II)

501 SOCIAL AND BEHAVIORAL SCIENCES
502 GEOPHYSICS
503 GEOLOGY
504 OCEANOGRAPHY
505 PHYSICS
506 MEDICAL SCIENCES
507 CHEMISTRY
508 BIOLOGY
509 ASTRONOMY AND CELESTIAL NAVIGATION
510 PETROLEUM ENGINEERING
511 HYDRAULIC ENGINEERING
512 NUCLEAR ENGINEERING
513 ELECTRICAL ENGINEERING
514 MECHANICAL ENGINEERING
515 CIVIL ENGINEERING
516 CHEMICAL ENGINEERING
517 AERONAUTICAL ENGINEERING
518 STRUCTURAL ENGINEERING
519 SYSTEM THEORY

600 MANAGEMENT SCIENCES AND OPERATIONS RESEARCH (VOLUME III)

602 PERT
603 CRITICAL PATH ANALYSIS
604 OPTIMIZATION PROGRAMS
605 LINEAR PROGRAMMING
606 DISCRETE SYSTEMS SIMULATION
607 CONTINUOUS SYSTEMS SIMULATION
608 FORECASTING TECHNIQUES
610 DYNAMIC PROGRAMMING

700 BUSINESS AND MANUFACTURING APPLICATIONS (VOLUME III)

701 JOB REPORTING
702 QUALITY ASSURANCE PERFORMANCE ANALYSIS
703 QUALITY ASSURANCE TESTING
704 NUMERICAL CONTROL
705 BILL OF MATERIALS
706 PAYROLL ACCOUNTING
707 WORK-IN-PROCESS CONTROL
708 INVENTORY ANALYSIS
709 ACCOUNTS PAYABLE
710 SALES FORECASTING
711 ACCOUNTS RECEIVABLE
712 FINANCIAL ANALYSIS
713 INVESTMENT ANALYSIS
714 ECONOMIC ANALYSIS
716 BUDGETING PROGRAMS
717 BUSINESS INFORMATION SYSTEMS
718 BUSINESS SERVICES

800 EDUCATION (VOLUME IV)

801 MATHEMATICS (EDUCATION)
810 PROGRAMMING AND COMPUTER SCIENCE (EDUCATION)
820 ENGINEERING (EDUCATION)
830 ECONOMICS (EDUCATION)
833 SCIENCE (EDUCATION)
850 FINE ARTS (EDUCATION)
860 SOCIAL SCIENCE (EDUCATION)
863 HISTORY (EDUCATION)
870 ENGLISH (EDUCATION)
871 FOREIGN LANGUAGES (EDUCATION)
872 READING (EDUCATION)
880 BUSINESS (EDUCATION)
885 EDUCATIONAL ADMINISTRATION
890 VOCATIONAL (EDUCATION)

900 UNCLASSIFIED (VOLUME V)

903 GAMES

INTRODUCTION

GENERAL

HP designs, manufactures and markets more than 3600 products, including electronic test and measuring instruments and systems; computational products that include desk top and personal-sized calculators, minicomputers and computer systems used in science, education, business and industry, medical electronic products for patient monitoring, diagnosis, and research; chromatographic and spectroscopic instrumentation for chemical analysis; and a variety of solid-state components.

Corporate, International, and Intercontinental Operations headquarters and the corporate research are located in Palo Alto, California; European Operations headquarters are in Geneva, Switzerland. HP has sales and service facilities in 65 countries.

THE HP CONTRIBUTED SOFTWARE CENTER

Hewlett-Packard's General Systems Division makes available to all HP 2000 and HP 3000 system users a wide variety of computer programs through the HP Contributed Software Center. The Contributed Software Center is composed of the General System Division's two contributed libraries; the 2000 Series (BASIC) and the 3000 Series. The Center serves as the administrator for the libraries. Software is submitted to the Center which then prepares it for distribution. The preparation includes indexing programs according to their use or function, and publishing library catalogs and handbooks which contain abstracts and/or documentation.

Contributed software is written by users of HP systems and submitted to the Center for inclusion in the appropriate library. These programs range from file manipulation routines to educational packages and apply to several different HP systems. Before writing a particular application scan the catalogs or handbooks containing information on programs written for the system you are using. Some programs can be used without modification while other programs serve as a starting point for developing special purpose software.

New programs are welcome for consideration as entries to the HP 2000 Series, and the HP 3000 Series Contributed Library. It is HP's opportunity to expand communication among HP computer system users. Minimum submittal requirements are (1) machine readable source paper or magnetic tape (documentation should be contained in the code, when possible), (2) a typed and reporducible program documentation form (these forms are printed in contributed program catalogs and are also avaiiable on request from the Center). All program packages should be wrapped securely and sent to:

Hewlett-Packard Contributed Software Center
General Systems Division
5303 Stevens Creek Blvd.
Santa Clara, Calif. 95050

Contributed software is checked by HP personnel; however, it is impractical to test programs under all circumstances. HEWLETT-PACKARD MAKES NO WARRANTY EXPRESSED OR IMPLIED AND ASSUMES NO RESPONSIBILITY IN CONNECTION WITH THE CONTRIBUTED PROGRAM MATERIAL. However, if you encounter an error, software report forms are supplied with library handbooks and catalogs. Fill them out and forward them to the Center. We will in turn direct them to the contributor of the software.

2000 SERIES (BASIC)

Program written for the HP 2000 Systems are documented in 5 Volumes, an addendum to Volumes I–IV, plus additional extended documentation for certain individual programs.

3000 SERIES

Programs written for HP 3000 Systems are abstracted in a Contributed Software Index and Catalog. The library is available as a complete package containing the Index and Catalog, extended documentation, and a corresponding magnetic tape.

NEW ORGANIZATION OF LIBRARY

The HP 2000 Series Contrubuted Library consists of the five volumes and addendum documentation for the former 2000F Level Library, plus manual updates and one 2400' reel of magnetic tape. The manual updates accumulate all changes to the 2000F documentation which relate to the newest system in the 2000 SERIES BASIC family. The magnetic tape contains all of the software from the 2000F Contributed Library arranged in twelve separate accounts—six (ZXXX's), and six (CXXX's). The "Z" accounts range from Z901 which corresponds to the software and documentation from Volume 1, to Z906 which corresponds to the software and documentation from the addendum. The programs which reside in the "Z" accounts have been *tested, unrestricted,* and *will execute on the new computer system.* The "C" accounts range from C901 which corresponds to software from Volume 1 to C906 which corresponds to the software from the addendum. These programs have also been tested but will not execute on the new computer system without user modification. The Contributed Software Center is not recoding the "C" account programs. Note: There is no C905 account; all of the games will execute on the new system.

Program documentation is arranged alphabetically, by calling Name, within each major category. Each volume represents a particular catagory or categories. The addendum Volume updates Volumes I–IV.

VOLUMES

VOLUME I	(100)	DATA HANDLING
	(200)	TESTING, DEBUGGING AND PROGRAMMING AIDS
VOLUME II	(300)	MATH AND NUMBERICAL ANALYSIS
	(400)	PROBABILITY AND STATISTICS
	(500)	SCIENTIFIC AND ENGINEERING APPLICATIONS
VOLUME III	(600)	MANAGEMENT SCIENCES AND OPERATIONS RESEARCH
	(700)	BUSINESS AND MANUFACTURING APPLICATIONS
VOLUME IV	(800)	EDUCATION
VOLUME V	(900)	MISCELLANEOUS (GAMES)**

Plotting routines previously classified under 904 are now found in Volume I under DATA HANDLING; this leaves Volume V exclusively for GAMES.

ORDERING INFORMATION

Contact your local HP Sales Office for ordering information on Contributed Software.

There are (4) four ways to order the library.

1. SOFTWARE AND DOCUMENTATION

HP 36600A (800 BPI)	HP 2000 Series Mag Tap of software and 5 Volumes of documetation plus the addendum to Volumes I–IV
HP 36600A-option 100 (1600 BPI)	HP 2000 Series Mag Tape of software and 5 Volumes of documentation plus addendum to Volumes I–IV.

2. SOFTWARE

HP 36600 -10001 (800 BPI)	HP 2000 Series MAG Tape of software
HP 36000 -11001 (1600 BPI)	HP 2000 Series Mag Tape of software

3. DOCUMENTATION (Collection)

HP 36600 -90001	5 Volumes of documentation plus the addendum documentation

4. DOCUMENTATION

HP 36000-91001	Volume I	HP 2000 BASIC Program Library
HP 36000-91002	Volume II	HP 2000 BASIC Program Library
HP 36000-91003	Volume III	HP 2000 BASIC Program Library
HP 36000-91004	Volume IV	HP 2000 BASIC Program Library
HP 36000-91005	Volume V	HP 2000 BASIC Program Library
HP 36000-920001	Addendum to Volumes I-IV	HP 2000 BASIC Program Library

EXTENDED DOCUMENTATION

FINDIT Users Manual	36250, Option DOO
CTC1 Documentation	36210, Option DOO
CTC2 Documentation	36311, Option DOO
CTC3 Documentation	36212, Option DOO
CTC4 Documentation	36213, Option DOO
CTC5 Documentation	36214, Option DOO
CTC6 Documentation	36638, Option DOO
TSBILL Documentation	36888-90039
BASP Documentation	36888-90022
MUSIC	36888-90028

ADDITIONAL ORDERING INFORMATION

If you are upgrading from a 2000F to the new 2000 Series System, manual updates are separately available by sending your request to:

Software/Publications Distribution
Hewlett Packard Company
5303 Stevens Creek Blvd.
Santa Clara, Calif. 95050

Please give the name of the manual, it's part number, and state that the update is required, not the complete manual. There is no charge for the manual updates.

For Example, to order Volume 1 update request:

HP 2000 Series Contributed Library, Vol 1.

Part Number 36000-91001
Update Only

ERRORS IN CONTRIBUTED SOFTWARE

Every HP BASIC Program included in the Contributed Library is checked by HP personnel and verified for accuracy with the sample RUN submitted. However, it is impractical to test programs under all circumstances, and HP does not assume responsibility for errors in contributed software. If you do encounter errors, please report them to the HP Contributed Software Center on the Error Report form supplied with this publication.

SYSTEM SPECIFICATIONS

Library programs have been collected over a period of years, and some of the earlier programs were written for a "single terminal" BASIC system, or an early version of the HP 2000 series Time-Share systems.

The chart below lists varying system features. In many cases slight modifications in coding will allow a program to RUN on systems other than the one for which it was originally written.

Program Features	2000A	2000B	2000C	2000F	2000C High Speed 2000I	2000 Series BASIC
Maximum program size	5100 words	5100 words	10,000 words	4140 words	10,000 words	10,000 words
Maximum No. of files	8	8	16	4	16	16
Maximum Number of Record/File	128	128	32,767	48	32,767	32,767
Maximum Number of Words/Record	64	64	256	128	256	256
Programmable Functions						
TIME		X	X	X	X	X
ENTER		X	X		X	X
COMMON		X	X	X	X	X
CHAIN		Chain-"name" X	Chain-$name stmt number	Chain-$name X	chain-$name stmt number	chain-"name" X
PRINT USING (IMAGE)			X		X	X
BRK					X	programmably detectable X
ASSIGN			X		X	X
RESTARTABLE RND			X	X	X	X
SPACE			X		X	X
LINE			X		X	X
Additional functions on 2000 Series BASIC						
ABS,ATN,CHR$,CONS,COS,						X
CTL,EXP,IDN,INT,LEN,						X
LOG,NUM,POS,REC,SGN,						X
SIN,SPA,SQR,SYS,						X
TAB,TAN,TIM,TRN,						X
TYP,UPS$,ZER						X

RELATED INFORMATION

EDUCATIONAL USER'S GROUP AT HP

The HP Educational User's Group is a worldwide organization of people sharing similar ideas, goals and concerns about education computing. The continuing focus of the User's Group is the exchange of ideas and experiences, channeled through periodic all-user meetings, regional sub-group activities and the Educational Newsletter.

For more information on these activities, contact: Educational User Services, Hewlett-Packard Company, 5303 Stevens Creek Blvd., Santa Clara, California 95050.

THE HP CLEARING HOUSE

The HP Clearinghouse was established in January, 1975 as an attempt to bring under one cover all those computer applications that would be of potential interest to HP users. The first catalog was printed in June, 1975 and contains information on some 200 applications, approximately 100 of them submitted by users. The catalog is organized into four categories: (1) Instructional Applications (presented by subject area); (2) Administrative Applications (listed by application type, e.g. student information systems); (3) Educational Utility Packages (CAI authoring/execution languages, IDF utilities, etc.); and (4) References (books, periodicals, and bibliographies). There are also six cross-reference indexes. This catalog is updated at approximately six-month intervals. The Clearinghouse disseminates information only – actual software is distributed by the originator or through the HP 2000 Series Contributed Library.

There are a number of manuals and documents relating to the HP 2000 Series Basic System that may be helpful to you.

2000/F to 2000/Access System Upgrade Kit and Conversion Program Manual (19665-90001)

2000/F to 2000 Access System Educational Application Upgrades (19665-90002)

Access BASIC Reference Manual, HP 2000 (22687-90001)

Access Operator's Manual, HP 2000 (22687-90005)

Access System Operator's Pocket Guide (22687-9007)

College Information System – System Overview (24384-90001)

College Information System Reference Manual (24384-90003)

College Information System – Technical Manual (24384-90005)

Course Writing Facility Reference Manual (22692-90001)

FCOPY/2000 Reference Manual (22700-90001)

HP MATH for HP 2000 Access Curriculum Guide (22693-90003)

HP MATH for HP 2000 Access Proctor's Manual (22693-90002)

HP MATH for HP 2000 Access Teacher's Handbook (22693-90001)

Instructional Dialogue Facility for HP 2000 Access Author's Manual (22691-90003)

Instructional Dialogue Facility for HP 2000 Access Author's Pocket Guide (22691-90004)

Instructional Dialogue Facility for HP 2000 Access Course Developer's (22691-90002)

Instructional Dialogue Facility for HP 2000 Access Proctor's Manual (22691-90001)

Instructional Management Facility for HP 2000 Access Proctor's Manual (22690-90001)

Instructional Management Facility for HP 2000 Access System Manager's Reference Manual (22690-90002)

Learning Timeshare BASIC (22687-90009)

Telecommunications Supervisory Package/2000 Manager's Manual (20240-90001)

Telecommunications Supervisory Package/2000 User's Manual (20240-90002)

GENERAL

Hewlett-Packard is a major designer and manufacturer of electronics for measurement, analysis and computation. HP customers in science, industry, medicine, and education know and appreciate Hewlett-Packard's reputation for technical excellence, quality, and reliability.

Over 170 world-wide offices sell and service the products of 21 manufacturing facilities located in the United States, Europe, and the Far East.

THE HP 2000 CONTRIBUTED LIBRARY

Hewlett-Packard makes available to all users a wide variety of computer programs through the HP 2000 Contributed Library.

Before writing a program for your particular application, scan the list of contributed programs. (A complete Index of contributed programs is available at your local HP sales office). You may be able to use these programs without modification, or as a starting point for developing your own special-purpose software.

The Contributed Library collects, indexes and distributes programs submitted by HP users throughout the world. These programs range from complex data communications packages to educational games, and all are classified according to the functions they perform.

2000 BASIC

Programs written in HP 2000 BASIC are documented in 5 volumes, plus additional user manuals for certain individual programs.

2000 NON-BASIC

Programs written for the HP 2000 series computers in FORTRAN, ALGOL, HP Assembly language, etc. are abstracted in the HP Program Catalog available from your local HP sales office. This catalog contains a number of programs for use with HP Time-Sharing systems, providing conversion capabilities, diagnostics, etc.

NEW ORGANIZATION OF LIBRARY

Because of the rapid growth of library contributions, it has been necessary to place a new emphasis on including only programs of very widespread usefulness. A Program Review Committee screens new submittals to determine this particular feature. Also, a number of programs have been purged from the library, where it was decided that a widespread application did not exist. You may elect to retain the documentation or software for one of these programs; however, HP will not be reprinting or updating them.

The documentation for BASIC Library programs has been completely reprinted and reorganized. There are five volumes available, and programs are arranged alphabetically, by calling NAMe, within each major category.

Volume I	(100) DATA HANDLING (200) TESTING, DEBUGGING AND PROGRAMMING AIDS
Volume II	(300) MATH AND NUMERICAL ANALYSIS (400) PROBABILITY AND STATISTICS (500) SCIENTIFIC AND ENGINEERING APPLICATIONS
Volume III	(600) MANAGEMENT SCIENCES AND OPERATIONS RESEARCH (700) BUSINESS AND MANUFACTURING APPLICATIONS
Volume IV	(800) EDUCATION
Volume V	(900) MISCELLANEOUS (GAMES) **

** Plotting routines previously classified under 904 are now found in Volume I under DATA HANDLING. This leaves Volume V exclusively for GAMES.

ORDERING INFORMATION

Contact your local HP sales office for ordering information of contributed software. Programs are available individually on paper tape, or collectively, on magnetic tape. Documentation is provided in the 5 volumes of BASIC Handbooks, and in some cases additional user manuals and classroom supplementary materials are available. (See list of Supplementary Documentation).

DOCUMENTATION

Volume I	HP 36000-91001	HP BASIC Program Library (100,200)
Volume II	HP 36000-91002	HP BASIC Program Library (300,400,500)
Volume III	HP 36000-91003	HP BASIC Program Library (600,700)
Volume IV	HP 36000-91004	HP BASIC Program Library (800)
Volume V	HP 36000-91005	HP BASIC Program Library (900) (GAMES)

SOFTWARE (HP 2000C'/F MAG TAPE DUMP)

*	HP 36000-10001	HP BASIC Contributed Software (100,200)
*	HP 36000-10002	HP BASIC Contributed Software (300,400,500)
*	HP 36000-10003	HP BASIC Contributed Software (600,700)
*	HP 36000-10004	HP BASIC Contributed Software (800)
*	HP 36000-10005	HP BASIC Contributed Software (900) (GAMES)

* 800 BPI. (1600 BPI mag tapes are also available under separate order number)

SUPPLEMENTARY DOCUMENTATION

FINDIT Users Manual	36250, Option D00
CTC1 Documentation	36210, Option D00
CTC2 Documentation	36211, Option D00
CTC3 Documentation	36212, Option D00
CTC4 Documentation	36213, Option D00
CTC5 Documentation	36214, Option D00
CTC6 Documentation	36638, Option D00
PILOT Users Manual	5951-5660
COBOL/2000 Primer	5951-5664
IDA	5951-5606
GRAZE (Student Manual)	5951-5653
(Teacher's Guide)	5951-5654
(Classroom Set)	5951-5655
CASE1	5951-5661
CASE2	5951-5662

UPDATES

The BASIC Library will be updated every 6 months. An addendum is printed, containing all new and revised programs in loose-leaf, 3-hole punched format to be easily added to your handbooks. A new Index is also published at this time to announce the release of new addenda and provide a complete updated list of library programs. Additions and revisions are flagged for your reference. Again, contact your local HP sales office to order addenda or a new Index.

SYSTEMS SPECIFICATIONS

Library programs have been collected over a period of years, and some of the earlier programs were written for a "single terminal" BASIC system, or an early version of the HP 2000 series Time-Share systems.

The chart below lists varying system features. In many cases slight modifications in coding will allow a program to RUN on systems other than the one for which it was originally written. The Index listing all Library programs indicates system compatibility for individual programs.

Program Features	2000A	2000B	2000C	2000E	2000C High-Speed 2000F
Maximum Program Size	5100 Words	5100 Words	10000 Words	4180 Words	10000 Words
Maximum Number of Files	8	8	16	4	16
Maximum Number of Records/File	128	128	32767	48	32767
Maximum Number of Words/Record	64	64	256	128	256
Programmable Functions:					
TIME		X	X	X	X
ENTER		X	X		X
COMMON		X	X	X	X
CHAINS		Chain-"Name"	Chain-$Name Statement No.	Chain-$Name	Chain-$Name Statement No.
		X	X	X	X
PRINT USING (IMAGE)			X		X
BRK					X
ASSIGN			X		X
RESTARTABLE RND			X	X	X
SPACE			X		X
LINE			X		X

ERRORS IN CONTRIBUTED SOFTWARE

Every HP BASIC Program included in the Contributed Library is checked by HP personnel and verified for accuracy with the sample RUN submitted. However, it is impractical to test programs under all circumstances, and HP does not assume responsibility for errors in contributed software. If you do encounter errors, please report them to the HP Contributed Library on the Error Report form supplied with this publication.

RELATED INFORMATION

An active Educational Users' Group at HP invites inquiries. Also, Hewlett-Packard offers a number of supported programs in Education Administration and Instruction. For more information on these activities, contact the Education Marketing Department, Hewlett-Packard Company, 11000 Wolfe Road, Cupertino, California 95014.

There are a number of manuals and documents relating to HP 2000 series Time-Sharing Systems that may be useful to you:

LANGUAGE MANUALS:

A Guide to HP Educational Basic (02116-91773)
HP BASIC (02116-9077)
2000F: Time-Shared BASIC Programmers' Guide (02000-90073)

OPERATING SYSTEM MANUALS:

2000F: Time-Shared BASIC Operator's Guide (02000-90074)

EDUCATIONAL APPLICATIONS MANUALS:

2000C/2000F System Operator Instructions for Educational Application (02000-90046)

2000C/2000F Instructional Management Facility and Instructional Dialogue Facility—Proctors Manual (02000-90047)

2000C/2000F Mathematics Drill and Practice Program—Proctors Manual (02000-90051)

2000C/2000F Instructional Dialogue Facility—Authors Manual (02000-90055)

2000C/2000F IDF Author's Pocket Guide (02000-90076)

2000C/2000F Mathematics Drill and Practice Program—Teachers Handbook (02000-90052)

COPYFL (02000-90032)

EDCALC (02000-90033)

Integer to String (02000-90035)

Date and Time (02000-90036)

2000C/2000F Introduction to Mathematics Drill and Practice (02000-90050)

2000C/2000F Mathematics Drill and Practice Curriculum Guide (02000-90053)

Course Developers' Manual for IDF-1 and IMF-1 (02000-90061)

Upshift (02000-90037)

Character Removal (02000-90038)

Key Word Search (02000-90039)

Downshift (02000-90040)

String Match with "Don't Cares" (02000-90041)

String to Number (02000-90042)

Student Response Analysis (02000-90043)

The preceding publications are available at nominal cost through your local HP sales office.

HEWLETT-PACKARD CONTRIBUTED SOFTWARE CENTER
DOCUMENTATION FORM FOR CONTRIBUTED BASIC PROGRAMS

TITLE

PROGRAM NAME CLASSIFICATION CODE ☐☐☐

SELECT UP TO FOUR CROSS REFERENCE WORDS FROM CROSS REFERENCE INDEX _____ _____ _____ _____

DESCRIPTION ( ) Program ( ) Subroutine

(Please include the specific application of your program – i.e., how do *you* use it, or recommend its application.)

USER INSTRUCTIONS

If possible, please include 'INSTRUCTIONS' as an option in your program. (Define the inputs requested by the program or subroutine. List the files used, and the data format of each. List the maximum file size. If applicable, include algorithms used.)

NOTE ON SUBROUTINES: The following conventions have been adopted for stand-alone subroutines. Variable names should begin with Z. When more than 10 variables are used, Z, . . . Z9, list the other variable names under Special Considerations. Subroutine line number should begin at 9000.

SYSTEM SPECIFICATIONS

System: ( ) Single Terminal Basic ( ) 2000A ( ) 2000B ( ) 2000C ( ) 2000E ( ) 2000C'/F ( ) 2000 Series

Terminal: ( ) Teletype ( ) Mark Sense Card Reader ( ) CRT ( ) Other _____

Note: Does this program use the BRK function? ( ) Yes ( ) No

SPECIAL CONSIDERATIONS

List any special hardware requirements, subroutine variable names not beginning with a 'Z', accuracy limitations, literature references, etc.

CONTRIBUTOR'S NAME AND ORGANIZATION ADDRESS

TO BE PUBLISHED? ( ) yes ( ) no

DISCLAIMER

To the best of my knowledge this contributed program is free of any proprietary information and I hereby agree that HP may reproduce, publish, and use it, and authorize others to do so without liability of any kind.

Signature _____ Date _____

Attach a sample run including input data and resulting TTY output data. Send a paper tape, or whenever possible, please send program on 2000 Series dump tape, ID C915.

Do you use this program for instructional purposes?

What age level are the students?

Please briefly describe the course, and topics within the course.

ERROR REPORT FORM
(HP BASIC CONTRIBUTED)

Comment fully on any software "bugs" in the space provided and enclose any teleprinter output that may be useful in defining the problem. A copy will be forwarded to the contributor. A reply will be returned to the person who submits this report. Send completed report to:

Hewlett-Packard Company
HP2000 Series Contributed Library
5303 Stevens Creek Blvd.
Santa Clara, California 95050

Submitted By — Date

Organization Name — Program Name

Address — Program No.

City, State, Zip

Phone

Has software been modified by user? NO YES (If YES, explain below)

Enclosed References:

TTY LOG LISTING Corrected Tape Corrected LISTING

VOLUME I
CONTENTS

100 DATA HANDLING

VOLUME I
CONTENTS

100 DATA HANDLING Continued

200 TESTING, DEBUGGING AND PROGRAMMING AIDS

VOLUME I
CONTENTS

100 DATA HANDLING

VOLUME I
CONTENTS (Continued)

100 DATA HANDLING (Continued)

200 TESTING, DEBUGGING AND PROGRAMMING AIDS

CONTRIBUTED PROGRAM **BASIC**

~~2~~ASCII
36257

TITLE: ASCII CODE GENERATOR

DESCRIPTION: This program generates an ASCII tape and code sheet.

INSTRUCTIONS: Run the program. Each time it stops at an enter statement, tear off the paper, then push return.

The program will generate a four page ASCII tape code sheet.

SPECIAL CONSIDERATIONS: The file "ASCII" must be present in the library, and have been set up by the program ~~ASCII*~~ ASCII2, HP 36256A.

The Aardvark and Company Writing Team has designed programs to take up an absolute minimum of computer storage and perform a maximum purpose. The team encourages people to send good programs to Aardvark. As a slight encouragement, the team will give anyone who sends a program which is accepted a free "subscription" to the program handbook, and include the contributor as a member of the writing team.

ACKNOWLEDGEMENTS: Aardvark and Company
2130 Bell Court
Lakewood, Colorado 80215

RUN

```
RUN
?ASCII

THIS PROGRAM GENERATES AN ASCII CODE SHEET.
PLEASE TEAR OFF YOUR PAPER AND PUSH  RETURN
```

ASCII CODE
●=1 OR HOLE Ø=Ø OR NOT HOLE .=GUIDE HOLE

ØØØØØ.ØØØ	CTRL @	W.O. PARITY		ØØ●ØØ.ØØØ	[SPACE]	W.O. PARITY	
ØØØØØ.ØØ●	CTRL A	W.O. PARITY		ØØ●ØØ.ØØ●	!	W.O. PARITY	
ØØØØØ.Ø●Ø	CTRL B	W.O. PARITY		ØØ●ØØ.Ø●Ø	"	W.O. PARITY	
ØØØØØ.Ø●●	CTRL C	W.O. PARITY		ØØ●ØØ.Ø●●	#	W.O. PARITY	
ØØØØØ.●ØØ	CTRL D	W.O. PARITY		ØØ●ØØ.●ØØ	$	W.O. PARITY	
ØØØØØ.●Ø●	CTRL E	W.O. PARITY		ØØ●ØØ.●Ø●	%	W.O. PARITY	
ØØØØØ.●●Ø	CTRL F	W.O. PARITY		ØØ●ØØ.●●Ø	&	W.O. PARITY	
ØØØØØ.●●●	CTRL G	W.O. PARITY		ØØ●ØØ.●●●	'	W.O. PARITY	
ØØØØ●.ØØØ	CTRL H	W.O. PARITY		ØØ●Ø●.ØØØ	(	W.O. PARITY	
ØØØØ●.ØØ●	CTRL I	W.O. PARITY		ØØ●Ø●.ØØ●	)	W.O. PARITY	
ØØØØ●.Ø●Ø	CTRL J	W.O. PARITY		ØØ●Ø●.Ø●Ø	*	W.O. PARITY	
ØØØØ●.Ø●●	CTRL K	W.O. PARITY		ØØ●Ø●.Ø●●	+	W.O. PARITY	
ØØØØ●.●ØØ	CTRL L	W.O. PARITY		ØØ●Ø●.●ØØ	,	W.O. PARITY	
ØØØØ●.●Ø●	CTRL M	W.O. PARITY		ØØ●Ø●.●Ø●	-	W.O. PARITY	
ØØØØ●.●●Ø	CTRL N	W.O. PARITY		ØØ●Ø●.●●Ø	.	W.O. PARITY	
ØØØØ●.●●●	CTRL O	W.O. PARITY		ØØ●Ø●.●●●	/	W.O. PARITY	
ØØØ●Ø.ØØØ	CTRL P	W.O. PARITY		ØØ●●Ø.ØØØ	Ø	W.O. PARITY	
ØØØ●Ø.ØØ●	CTRL Q	W.O. PARITY		ØØ●●Ø.ØØ●	1	W.O. PARITY	
ØØØ●Ø.Ø●Ø	CTRL R	W.O. PARITY		ØØ●●Ø.Ø●Ø	2	W.O. PARITY	
ØØØ●Ø.Ø●●	CTRL S	W.O. PARITY		ØØ●●Ø.Ø●●	3	W.O. PARITY	
ØØØ●Ø.●ØØ	CTRL T	W.O. PARITY		ØØ●●Ø.●ØØ	4	W.O. PARITY	
ØØØ●Ø.●Ø●	CTRL U	W.O. PARITY		ØØ●●Ø.●Ø●	5	W.O. PARITY	
ØØØ●Ø.●●Ø	CTRL V	W.O. PARITY		ØØ●●Ø.●●Ø	6	W.O. PARITY	
ØØØ●Ø.●●●	CTRL W	W.O. PARITY		ØØ●●Ø.●●●	7	W.O. PARITY	
ØØØ●●.ØØØ	CTRL X	W.O. PARITY		ØØ●●●.ØØØ	8	W.O. PARITY	
ØØØ●●.ØØ●	CTRL Y	W.O. PARITY		ØØ●●●.ØØ●	9	W.O. PARITY	
ØØØ●●.Ø●Ø	CTRL Z	W.O. PARITY		ØØ●●●.Ø●Ø	:	W.O. PARITY	
ØØØ●●.Ø●●	CTRL [	W.O. PARITY		ØØ●●●.Ø●●	;	W.O. PARITY	
ØØØ●●.●ØØ	CTRL \	W.O. PARITY		ØØ●●●.●ØØ	<	W.O. PARITY	
ØØØ●●.●Ø●	CTRL]	W.O. PARITY		ØØ●●●.●Ø●	=	W.O. PARITY	
ØØØ●●.●●Ø	CTRL ↑	W.O. PARITY		ØØ●●●.●●Ø	>	W.O. PARITY	
ØØØ●●.●●●	CTRL ←	W.O. PARITY		ØØ●●●.●●●	?	W.O. PARITY	

ASCII CODE
●=1 OR HOLE Ø=Ø OR NOT HOLE .=GUIDE HOLE

Ø●ØØØ.ØØØ	@	W.O. PARITY		Ø●●ØØ.ØØØ	LOWER CASE @	W.O. PARITY	
Ø●ØØØ.ØØ●	A	W.O. PARITY		Ø●●ØØ.ØØ●	LOWER CASE A	W.O. PARITY	
Ø●ØØØ.Ø●Ø	B	W.O. PARITY		Ø●●ØØ.Ø●Ø	LOWER CASE B	W.O. PARITY	
Ø●ØØØ.Ø●●	C	W.O. PARITY		Ø●●ØØ.Ø●●	LOWER CASE C	W.O. PARITY	
Ø●ØØØ.●ØØ	D	W.O. PARITY		Ø●●ØØ.●ØØ	LOWER CASE D	W.O. PARITY	
Ø●ØØØ.●Ø●	E	W.O. PARITY		Ø●●ØØ.●Ø●	LOWER CASE E	W.O. PARITY	
Ø●ØØØ.●●Ø	F	W.O. PARITY		Ø●●ØØ.●●Ø	LOWER CASE F	W.O. PARITY	
Ø●ØØØ.●●●	G	W.O. PARITY		Ø●●ØØ.●●●	LOWER CASE G	W.O. PARITY	
Ø●Ø●Ø.ØØØ	H	W.O. PARITY		Ø●●Ø●.ØØØ	LOWER CASE H	W.O. PARITY	
Ø●Ø●Ø.ØØ●	I	W.O. PARITY		Ø●●Ø●.ØØ●	LOWER CASE I	W.O. PARITY	
Ø●Ø●Ø.Ø●Ø	J	W.O. PARITY		Ø●●Ø●.Ø●Ø	LOWER CASE J	W.O. PARITY	
Ø●Ø●Ø.Ø●●	K	W.O. PARITY		Ø●●Ø●.Ø●●	LOWER CASE K	W.O. PARITY	
Ø●Ø●Ø.●ØØ	L	W.O. PARITY		Ø●●Ø●.●ØØ	LOWER CASE L	W.O. PARITY	
Ø●Ø●Ø.●Ø●	M	W.O. PARITY		Ø●●Ø●.●Ø●	LOWER CASE M	W.O. PARITY	
Ø●Ø●Ø.●●Ø	N	W.O. PARITY		Ø●●Ø●.●●Ø	LOWER CASE N	W.O. PARITY	
Ø●Ø●Ø.●●●	O	W.O. PARITY		Ø●●Ø●.●●●	LOWER CASE O	W.O. PARITY	
Ø●Ø●●.ØØØ	P	W.O. PARITY		Ø●●●Ø.ØØØ	LOWER CASE P	W.O. PARITY	
Ø●Ø●●.ØØ●	Q	W.O. PARITY		Ø●●●Ø.ØØ●	LOWER CASE Q	W.O. PARITY	
Ø●Ø●●.Ø●Ø	R	W.O. PARITY		Ø●●●Ø.Ø●Ø	LOWER CASE R	W.O. PARITY	
Ø●Ø●●.Ø●●	S	W.O. PARITY		Ø●●●Ø.Ø●●	LOWER CASE S	W.O. PARITY	
Ø●Ø●●.●ØØ	T	W.O. PARITY		Ø●●●Ø.●ØØ	LOWER CASE T	W.O. PARITY	
Ø●Ø●●.●Ø●	U	W.O. PARITY		Ø●●●Ø.●Ø●	LOWER CASE U	W.O. PARITY	
Ø●Ø●●.●●Ø	V	W.O. PARITY		Ø●●●Ø.●●Ø	LOWER CASE V	W.O. PARITY	
Ø●Ø●●.●●●	W	W.O. PARITY		Ø●●●Ø.●●●	LOWER CASE W	W.O. PARITY	

@@@@@.@@@	X	W.O. PARITY	@@@@@.@@@	LOWER CASE X	W.O. PARITY
@@@@@.@@@	Y	W.O. PARITY	@@@@@.@@@	LOWER CASE Y	W.O. PARITY
@@@@@.@@@	Z	W.O. PARITY	@@@@@.@@@	LOWER CASE Z	W.O. PARITY
@@@@@.@@@	[	W.O. PARITY	@@@@@.@@@	LOWER CASE [	W.O. PARITY
@@@@@.@@@	\	W.O. PARITY	@@@@@.@@@	LOWER CASE \	W.O. PARITY
@@@@@.@@@	]	W.O. PARITY	@@@@@.@@@	LOWER CASE]	W.O. PARITY
@@@@@.@@@	↑	W.O. PARITY	@@@@@.@@@	LOWER CASE ↑	W.O. PARITY
@@@@@.@@@	←	W.O. PARITY	@@@@@.@@@	LOWER CASE ←	W.O. PARITY

ASCII CODE
@=1 OR HOLE Ø=Ø OR NOT HOLE .=GUIDE HOLE

@@@@@.@@@	CTRL @	W. PARITY	@@@@@.@@@	[SPACE]	W. PARITY
@@@@@.@@@	CTRL A	W. PARITY	@@@@@.@@@	!	W. PARITY
@@@@@.@@@	CTRL B	W. PARITY	@@@@@.@@@	"	W. PARITY
@@@@@.@@@	CTRL C	W. PARITY	@@@@@.@@@	#	W. PARITY
@@@@@.@@@	CTRL D	W. PARITY	@@@@@.@@@	$	W. PARITY
@@@@@.@@@	CTRL E	W. PARITY	@@@@@.@@@	%	W. PARITY
@@@@@.@@@	CTRL F	W. PARITY	@@@@@.@@@	&	W. PARITY
@@@@@.@@@	CTRL G	W. PARITY	@@@@@.@@@	'	W. PARITY
@@@@@.@@@	CTRL H	W. PARITY	@@@@@.@@@	(	W. PARITY
@@@@@.@@@	CTRL I	W. PARITY	@@@@@.@@@	)	W. PARITY
@@@@@.@@@	CTRL J	W. PARITY	@@@@@.@@@	*	W. PARITY
@@@@@.@@@	CTRL K	W. PARITY	@@@@@.@@@	+	W. PARITY
@@@@@.@@@	CTRL L	W. PARITY	@@@@@.@@@	,	W. PARITY
@@@@@.@@@	CTRL M	W. PARITY	@@@@@.@@@	-	W. PARITY
@@@@@.@@@	CTRL N	W. PARITY	@@@@@.@@@	.	W. PARITY
@@@@@.@@@	CTRL O	W. PARITY	@@@@@.@@@	/	W. PARITY
@@@@@.@@@	CTRL P	W. PARITY	@@@@@.@@@	Ø	W. PARITY
@@@@@.@@@	CTRL Q	W. PARITY	@@@@@.@@@	1	W. PARITY
@@@@@.@@@	CTRL R	W. PARITY	@@@@@.@@@	2	W. PARITY
@@@@@.@@@	CTRL S	W. PARITY	@@@@@.@@@	3	W. PARITY
@@@@@.@@@	CTRL T	W. PARITY	@@@@@.@@@	4	W. PARITY
@@@@@.@@@	CTRL U	W. PARITY	@@@@@.@@@	5	W. PARITY
@@@@@.@@@	CTRL V	W. PARITY	@@@@@.@@@	6	W. PARITY
@@@@@.@@@	CTRL W	W. PARITY	@@@@@.@@@	7	W. PARITY
@@@@@.@@@	CTRL X	W. PARITY	@@@@@.@@@	8	W. PARITY
@@@@@.@@@	CTRL Y	W. PARITY	@@@@@.@@@	9	W. PARITY
@@@@@.@@@	CTRL Z	W. PARITY	@@@@@.@@@	:	W. PARITY
@@@@@.@@@	CTRL [	W. PARITY	@@@@@.@@@	;	W. PARITY
@@@@@.@@@	CTRL \	W. PARITY	@@@@@.@@@	<	W. PARITY
@@@@@.@@@	CTRL]	W. PARITY	@@@@@.@@@	=	W. PARITY
@@@@@.@@@	CTRL ↑	W. PARITY	@@@@@.@@@	>	W. PARITY
@@@@@.@@@	CTRL ←	W. PARITY	@@@@@.@@@	?	W. PARITY

ASCII CODE
@=1 OR HOLE Ø=Ø OR NOT HOLE .=GUIDE HOLE

@@@@@.@@@	@	W. PARITY	@@@@@.@@@	LOWER CASE @	W. PARITY
@@@@@.@@@	A	W. PARITY	@@@@@.@@@	LOWER CASE A	W. PARITY
@@@@@.@@@	B	W. PARITY	@@@@@.@@@	LOWER CASE B	W. PARITY
@@@@@.@@@	C	W. PARITY	@@@@@.@@@	LOWER CASE C	W. PARITY
@@@@@.@@@	D	W. PARITY	@@@@@.@@@	LOWER CASE D	W. PARITY
@@@@@.@@@	E	W. PARITY	@@@@@.@@@	LOWER CASE E	W. PARITY
@@@@@.@@@	F	W. PARITY	@@@@@.@@@	LOWER CASE F	W. PARITY
@@@@@.@@@	G	W. PARITY	@@@@@.@@@	LOWER CASE G	W. PARITY
@@@@@.@@@	H	W. PARITY	@@@@@.@@@	LOWER CASE H	W. PARITY
@@@@@.@@@	I	W. PARITY	@@@@@.@@@	LOWER CASE I	W. PARITY
@@@@@.@@@	J	W. PARITY	@@@@@.@@@	LOWER CASE J	W. PARITY
@@@@@.@@@	K	W. PARITY	@@@@@.@@@	LOWER CASE K	W. PARITY
@@@@@.@@@	L	W. PARITY	@@@@@.@@@	LOWER CASE L	W. PARITY
@@@@@.@@@	M	W. PARITY	@@@@@.@@@	LOWER CASE M	W. PARITY
@@@@@.@@@	N	W. PARITY	@@@@@.@@@	LOWER CASE N	W. PARITY
@@@@@.@@@	O	W. PARITY	@@@@@.@@@	LOWER CASE O	W. PARITY
@@@@@.@@@	P	W. PARITY	@@@@@.@@@	LOWER CASE P	W. PARITY
@@@@@.@@@	Q	W. PARITY	@@@@@.@@@	LOWER CASE Q	W. PARITY
@@@@@.@@@	R	W. PARITY	@@@@@.@@@	LOWER CASE R	W. PARITY
@@@@@.@@@	S	W. PARITY	@@@@@.@@@	LOWER CASE S	W. PARITY
@@@@@.@@@	T	W. PARITY	@@@@@.@@@	LOWER CASE T	W. PARITY
@@@@@.@@@	U	W. PARITY	@@@@@.@@@	LOWER CASE U	W. PARITY
@@@@@.@@@	V	W. PARITY	@@@@@.@@@	LOWER CASE V	W. PARITY
@@@@@.@@@	W	W. PARITY	@@@@@.@@@	LOWER CASE W	W. PARITY

88088.000	X	W. PARITY		88088.000	LOWER CASE X	W. PARITY
88088.008	Y	W. PARITY		88088.008	LOWER CASE Y	W. PARITY
88088.080	Z	W. PARITY		88088.080	LOWER CASE Z	W. PARITY
88088.088	[	W. PARITY		88088.088	LOWER CASE [	W. PARITY
88088.800	\	W. PARITY		88088.800	LOWER CASE \	W. PARITY
88088.808	]	W. PARITY		88088.808	LOWER CASE]	W. PARITY
88088.880	↑	W. PARITY		88088.880	LOWER CASE ↑	W. PARITY
88088.888	←	W. PARITY		88088.888	LOWER CASE ←	W. PARITY

CONTRIBUTED PROGRAM **BASIC**

ZAIRE
36807

TITLE: QUESTIONNAIRE ANALYSIS

DESCRIPTION:

The Questionnaire Analysis System is a collection of three computer programs (?INPUT, ?PRINT, and ?TABLE) which provide the capability of accepting and printing the results of a wide variety of questionnaires.

INSTRUCTIONS:

To use the system it is necessary to write a subprogram consisting of a series of DATA statements which contain the required information concerning the particular questionnaire. This subprogram is appended to ?INPUT which is used for entering the questionnaire responses and then to the program ?PRINT (and possibly also ?TABLE) which prints tabulations of the data. The latter programs permit the tabulation of data for selected subgroups as well as the entire population of respondants.

The Sample Questionnaire should be consulted for clarification of the sample RUNs.

The Subprogram.

When writing the subprogram, lines 9000, 9020, 9500, and 9999 must be present and of the form indicated below because these lines are explicitly referenced by the main programs. Line 9000 is a REM statement which identifies the subprogram.

Lines 9001 to 9019 contain four string data items: the coded data file name, the decoded data file name, the subgroup data file name, and the name of the questionnaire.

As data is entered in ?INPUT, they are stored in a coded form on the coded data file. Coding is performed to reduce the disk storage needed for the data. In general, about 8 responses will be stored in one number; thus a questionnaire with 80 questions would require about 10 numbers (20 words) per questionnaire. The exact number of words can be obtained by running the SIZ option in ?INPUT after the subprogram has been written (the file itself need not have been opened to run this option). This information together with the number of respondants will determine the size of the coded data file.

The decoded data file and the subgroup file need not exist (in which case some nonsense name such as NONE can be used). These files are used - if at all - in ?PRINT and ?TABLE. The decoded data file is desirable if many runs of ?PRINT are to be made because the decoding process is fairly time-consuming, but need be done only once if a decoded data file is available. The decoded data file requires one number for each question for each respondant and consequently is several times larger than the coded data file.

The subgroup file holds complete tabulations which can then be printed in ?TABLE in a format which permits easy comparisons between various subgroup response patterns. Its use also permits overnight computer runs to calculate the tabulations for several subgroups - a feature which avoids tying up a terminal during the day for possibly long non-printing periods.

The name of the questionnaire will be printed on each tabulation. Each question is described in a single data statement between line 9020 and 9499 inclusive. Such a statement contains three strings which contain the question name, the allowable responses, and the question description.

ACKNOWLEDGEMENTS: Dr. W. Y. Gateley,
Colorado College

INSTRUCTIONS (Continued)

NOTE: The major restriction is that each question response must be a single character or at least can be coded to a single character.

The question name is what one usually calls the question number, but we use the term "name" rather than "number" to emphasize that designations such as "1a" are acceptable - in fact any name may be used. The term "question number" will be used to identify the position of a question with the counting starting with the question described in line 9020. It is important to distinguish between a question's name and its number (although in many questionnaires they are identical); in the programs this is done by using $ to indicate a name and # to indicate a number. A question name must not exceed 7 characters.

The allowable responses must contain every possible (single character) response to a question and must include as the last possible response a space character which will indicate a non-response. The ?INPUT program will check each questionnaire entry and will not permit an unacceptable response to be recorded. Any ASCII character other than question mark (?) and quote (") may be used for an allowable response. The number of responses permitted for a given question may not exceed 72 characters.

The question description will be printed on each tabulation.

The question description DATA statements must be immediately followed by the statement

```
DATA Ø
```

which serves as a terminating flag (the line number of that statement must be less than 9500).

Line 9500 should read

```
9500  REM---SUBGROUPS---
```

Lines 9501 through 9998 contain the data necessary to produce subgroup tabulations. A "sub-group" is simply a subset of the population of all respondants. One subgroup which should always be defined is the entire population. Other subgroups will consist of those respondants who answered one or more questions in particular ways. For example, if the sex of the respondant were asked for in one question, we could tabulate the questionnaire responses of all males; if, in addition, a question asked for eye color, we could tabulate the responses of all blue-eyed females. The data necessary to describe a subgroup are (in the order in which they must appear in the DATA statement):

a. The name of the subgroup (a string)

b. The number of questions which are to be used to determine the subgroup (Ø for the all-respondant subgroup)

c. Then for each such question:

   the number (not name) of the question, the number of responses to that question to be used for selection, and a list of those response numbers (where the counting of response numbers starts from the beginning of the allowable response string for the particular question.

As many subgroups as are desired may be defined.

The last line in the subprogram must be

```
9999  END
```

The program must be named and saved. We suggest running ?INPUT (with the subprogram appended) and using the SIZ and QUE options before any questionnaire data are entered.

The ?INPUT Program.

?INPUT is used to enter the questionnaire data and may be successfully used by non-programmer personnel. After the RUN command is given, the computer types "OPTION:". The user may then select one of several options to be performed. After completing a requested option, the computer will again type "OPTION:". The available options are:

```
DON - to terminate the program (just a carriage return is equivalent to DON)
OPT - to list the options
ADD - to enter more questionnaires
FIX - to alter the responses in a previously entered questionnaire
REC - to list one or more entered records (a "record" is the set of all responses for one respondant)
QUE - to list all of the question data in the subprogram
SIZ - to list the values of certain parameters related to the particular questionnaire
DEL - to delete a particular record
```

The three-character options may be entered in either upper or lower case, but not mixed.

?INPUT should always be terminated by the use of the DON option. Use of the BREAK or control-C keys may result in the loss of data.

INSTRUCTIONS (Continued)

The ADD Option.

When the Add option is called, the computer counts the number of questionnaire records which have been previously entered and assigns the next integer to the next record. This number is printed at the terminal and should be written on the questionnaire for possible future reference.

The computer will then type "Enter from Q$ xx" where xx is the name of the first question. The user will then start entering the characters corresponding to the questionnaire answers.

At any point the user may push RETURN. When this is done, the computer will check the entered responses against the allowable ones. If an illegal response is detected, a request will be made for a new entry for that question. After the checking is completed, the computer will type "Enter from Q$ xx" where xx is the name of the next question, and the above process will continue until all responses have been entered.

The user may restart the entries at any Q$ by typing ? either in the entry string or in response to a request to fix a bad response; if this is done, the computer will list all of the record which as been successfully entered so far and then ask for the Q$ at which the user wishes to start.

If an excessive number of entries are made, the computer will inform the user and will simply delete the extra entries. Normally this will indicate an error on the user's part.

After all responses have been entered, the computer will ask "OK?". If the user is satisfied, he will respond with "YES", "Y", or simply RETURN; if he is not, he may type a ? which will cause the entire record to be printed and followed by another "OK?" or with a "NO" or "N" which will cause the computer to request the Q$s between which new entries are to be made.

After a record has been satisfactorily entered, the computer will go on to the next record. The ADD option is terminated by the user pushing just the RETURN key in response to "Enter from Q$ xx" (this can also be done even after some responses have been entered, in which case these responses will be ignored - that is, a partial record will not be written on the file).

The FIX Option.

The FIX option permits changing part or all of a particular record. The user will be asked for the desired record number and then the question names between which he wishes to make changes. From here on the operation is identical to that of the ADD option. After the changes have been made, the computer will ask for another record number; a RETURN at this point will terminate the option.

Output.

Printed output is obtained in either of two forms: a tabulation of a single subgroup or a tabulation which contains the result for up to 10 different subgroups. The latter is useful when comparisons between subgroups is desired. The format of the two forms can be seen in the sample RUN.

Regardless of the form to be used, ?PRINT must first be run (?TABLE is used only if the multi-subgroup printout is wanted).

The following options are available in ?PRINT:

DON - terminate program
OPT - to list the options
DEC - to decode the data
SUB - to print one or more single subgroup tabulations
PSG - write subgroup tabulations on the subgroup file for later use
FSG - same as SUB except the data is obtained from the subgroup file rather than the coded or decoded data files
LSG - list the subgroup descriptions

The LSG option should be run immediately to check the subgroup definitions.

If sufficient disk space is available for a decoded data file and if several runs are to be made and/or subgroups are to be processed, the DEC option should be used. The decoded data file requires one number for each question for each questionnaire. For example, a 50-question questionnaire given to 100 people will require a decoded data file holding 5000 numbers (hence 40 records). If the data is not decoded by the DEC option, decoding will be done for each tabulation; this can greatly increase the processing time in some instances.

INSTRUCTIONS (Continued)

The PSG option must be used if a multi-subgroup tabulation is desired (using ?TABLE), and may be used for single-subgroup tabulations. Its use is desirable in the latter case if lengthy processing is expected because the processing can be done overnight. Data written on the subgroup file during the execution of PSG consists of one number for each possible response plus two numbers for each subgroup (the total number of possible responses can be obtained from the SIZ option in ?INPUT). For example, if a questionnaire has a total of 200 possible responses and there are 5 subgroups, the subgroup file would need to hold (200 + 2) x 5 = 1010 numbers (hence, 8 records). Not all of the defined subgroups need be processed at one time, but it is preferable to do so unless disk space is not available.

Single subgroup tabulations are printed by using either the SUB or FSG options. The latter may be used only if PSG has been run previously.

The BREAK key may be used in ?PRINT without harming the coded data file. If it is used during either the DEC or PSG option, the option must be restarted.

The program ?TABLE has no options. It produces a tabulation for up to 10 subgroups (which have previously gone through the PSG option of ?PRINT). The subgroups may be selected in any order for the tabulation. The BREAK key may be used at any time.

All tabulations are printed in paged form and include the questionnaire name, the date, and page numbers.

In summary, for most cases we suggest that after the data has been entered, the user run (in this order) the ?PRINT options LSG, DEC, and PSG. The actual printouts can then be made without further computation by either the FSG option in ?PRINT or by ?TABLE, depending upon the type of tabulation desired.

SAMPLE QUESTIONNAIRE

Instructions: circle your answer to each question.

1(a). Sex? M F

1(b). Age?
A. under 20
B. 20 to 30
C. over 30

2. Political preference?
R: Republican
D: Democrat
I: Independent

XXX. Have you ever eaten Friendly Fred's Frankfurters? YES NO

Assume the responses from 10 people were as follows (a blank indicates no response):

Respondant	1(a)	1(b)	2	XXX
1	M	A	I	Y
2	F		D	N
3	F	C	R	N
4	F	A	I	Y
5	M	B		Y
6	M		D	
7	F	B	D	N
8	M	A	I	Y
9	M	C	I	Y
10	F	C	I	Y

```
RUN

NAM-?FRED
9000  REM---FRIENDLY FRED'S QUESTION DATA
9005  DATA "FREDCF","FREDDF","FREDSG"
9015  DATA "FRIENDLY FRED'S FRANKFURTERS"
9020  DATA "1(A)","MF ","SEX"
9030  DATA "1(B)","ABC ","AGE GROUP"
9040  DATA "2","RDI ","POLITICAL PREFERENCE"
9050  DATA "XXX","YN ","EATEN FFF"
9499  DATA 0
9500  REM---SUBGROUPS---
9505  DATA "ALL RESPONDANTS",0
9510  DATA "MALES",1,1,1,1
9520  DATA "INDEPENDENT FEMALES UNDER 20 OR OVER 30",3,1,1,2,2,2,1,3,3,3
9530  DATA "MALES OVER 30",2,1,1,1,2,1,3
9999  END
SAV
```

```
GET-?INPUT
APP-?FRED
RUN
?INPUT


OPTION:OPT
DON: EXIT PROGRAM (OR JUST USE CR)
OPT: LIST OPTIONS
ADD: ADD MORE DATA
FIX: CHANGE SOME DATA
QUE: LIST QUESTION DATA
REC: LIST DATA RECORDS
SIZ: PRINT SIZE DATA
DEL: DELETE A RECORD

OPTION:SIZ
# OF QUESTIONS: 4
TOTAL # OF RESPONSES: 14
MAX. # OF RESPONSES FOR A QUESTION: 4
RECORD SIZE FOR FILES (# OF NUMBERS): 1

OPTION:QUE
QUESTIONNAIRE: FRIENDLY FRED'S FRANKFURTERS
DATA FILE: FREDCF
DECODED DATA FILE: FREDDF
SUBGROUP FILE: FREDSG
Q#, RESPONSES, AND Q NAME ARE SEPARATED BY '//' IN LIST BELOW.)

  1.  1(A)//MF //SEX
  2.  1(B)//ABC //AGE GROUP
  3.  2//RDI //POLITICAL PREFERENCE
  4.  XXX//YN //EATEN FFF

OPTION:

DONE



OPE-FREDCF,2
OPE-FREDDF,2
OPE-FREDSG,2

RUN
?INPUT


OPTION:ADD

RECORD 1
ENTER FROM Q$ 1(A):MAIY
OK?

RECORD 2
ENTER FROM Q$ 1(A):FFDN
Q$ 1(B) BAD (F). RE-ENTER:
OK??
#  1:F DN//
OK?

RECORD 3
ENTER FROM Q$ 1(A):FCRN
OK?

RECORD 4
ENTER FROM Q$ 1(A):FAIY
OK?

RECORD 5
ENTER FROM Q$ 1(A):MB Y
OK?

RECORD 6
ENTER FROM Q$ 1(A):M D
OK?
```

```
RECORD 7
ENTER FROM Q$ 1(A):FFBDN
 1     EXTRA ENTRIES DELETED.
Q$ 1(B) BAD (F). RE-ENTER:?
#   1:F//
FROM Q$?1B
NO SUCH Q$. TRY AGAIN.
Q$?2
Q$ CANNOT BE GREATER THAN 1(B). TRY AGAIN.
FROM Q$?1(B)
ENTER FROM Q$ 1(B):BDN
 1     EXTRA ENTRIES DELETED.
OK?

RECORD 8
ENTER FROM Q$ 1(A):MAIY
OK?

RECORD 9
ENTER FROM Q$ 1(A):MBRY
OK?

RECORD 10
ENTER FROM Q$ 1(A):MCIY
OK?

RECORD 11
ENTER FROM Q$ 1(A):

OPTION:FIX
RECORD # (CR IF DONE):10
FROM Q$?2
  TO Q$?2
ENTER FROM Q$ 2:R
OK?
RECORD # (CR IF DONE):

OPTION:REC
STARTING AND ENDING RECORD #S?10,10

RECORD 10
#   1:MCRY//

OPTION:FIX
RECORD # (CR IF DONE):10
FROM Q$?2
  TO Q$?2
ENTER FROM Q$ 2:I
OK??
#   1:MCIY//
OK?NO
FROM Q$?1(A)
  TO Q$?1(A)
ENTER FROM Q$ 1(A):F
OK??
#   1:FCIY//
OK?
RECORD # (CR IF DONE):

OPTION:

DONE



GET-?PRINT
APP-?FRED
RUN
?PRINT


OPTION:OPT
DON: EXIT PROGRAM (OR JUST USE CR)
OPT: LIST OPTIONS
DEC: DECODE DATA, PUT INTO FILE 2
SUB: PRINT SUBGROUP ANALYSIS
PSG: PRINT SUBGROUP TABULATIONS ON FILE 3
FSG: GET SUBGROUP DATA FROM FILE 3 AND PRINT
LSG: LIST SUBGROUP DESCRIPTIONS
```

```
OPTION:LSG

SUBGROUPS:
 1    ALL RESPONDANTS
 2    MALES
        1:   1,
 3    INDEPENDENT FEMALES UNDER 20 OR OVER 30
        1:   2,
        2:   1,  3,
        3:   3,
 4    MALES OVER 30
        1:   1,
        2:   3,

OPTION:DEC
ALL DONE. 10   RECORDS WERE DECODED.

OPTION:PSG
STARTING & ENDING SUBGROUPS?1,4
SUBGROUP 1     DONE.  # OF RECORDS = 10
SUBGROUP 2     DONE.  # OF RECORDS = 5
SUBGROUP 3     DONE.  # OF RECORDS = 2
SUBGROUP 4     DONE.  # OF RECORDS = 0

OPTION:FSG
STARTING & ENDING SUBGROUPS?1,1
SET AT 1ST LINE OF PAGE & CR
```

FRIENDLY FRED'S FRANKFURTERS: ALL RESPONDANTS

DEC. 19, 1973; 10 RECORDS.

1(A) SEX:

ANS:	M	F	
#:	5	5	0
%:	50%	50%	0%

1(B) AGE GROUP:

ANS:	A	B	C	
#:	3	3	2	2
%:	30%	30%	20%	20%

2 POLITICAL PREFERENCE:

ANS:	R	D	I	
#:	2	3	4	1
%:	20%	30%	40%	10%

XXX EATEN FFF:

ANS:	Y	N	
#:	6	3	1
%:	60%	30%	10%

- P. 1 -

OPTION:

DONE

GET-?TABLE
APP-?FRED
RUN
?TABLE

TYPE SUBGROUP NUMBERS, ONE PER LINE. RETURN WHEN DONE.
? 1
? 2
? 3
?
SET AT TOP OF PAGE AND RETURN

SUBGROUP TABULATION: FRIENDLY FRED'S FRANKFURTERS

DEC. 19, 1973

SG#	# IN SG	SUBGROUP NAME
1	10	ALL RESPONDANTS
2	5	MALES
3	2	INDEPENDENT FEMALES UNDER 20 OR OVER 30

NOTE: THE SUBGROUP NUMBERS ARE LISTED ACROSS THE TOP OF EACH PAGE. THE POSSIBLE RESPONSES TO EACH QUESTION ARE PRINTED ON THE LEFT SIDE OF THE PAGE. FOR EACH RESPONSE AND EACH SUBGROUP IS GIVEN THE NUMBER OF RESPONSES, THE % OF ALL RESPONDANTS (ENCLOSED IN PARENTHESES), AND THE % OF ALL NON-BLANK RESPONDANTS (ENCLOSED IN BRACKETS).

- P. 1 -

Q$: 1(A) SEX

	1	2	3
M:	5	5	0
	(50)	(100)	(0)
	[50]	[100]	[0]
F:	5	0	2
	(50)	(0)	(100)
	[50]	[0]	[100]
:	0	0	0
	(0)	(0)	(0)

Q$: 1(B) AGE GROUP

	1	2	3
A:	3 (30) [37]	2 (40) [50]	1 (50) [50]
B:	3 (30) [37]	2 (40) [50]	0 (0) [0]
C:	2 (20) [25]	0 (0) [0]	1 (50) [50]
:	2 (20)	1 (20)	0 (0)

Q$: 2 POLITICAL PREFERENCE

	1	2	3
R:	2 (20) [22]	1 (20) [25]	0 (0) [0]
D:	3 (30) [33]	1 (20) [25]	0 (0) [0]
I:	4 (40) [44]	2 (40) [50]	2 (100) [100]
:	1 (10)	1 (20)	0 (0)

Q$: XXX EATEN FFF

	1	2	3
Y:	6 (60) [67]	4 (80) [100]	2 (100) [100]
N:	3 (30) [33]	0 (0) [0]	0 (0) [0]
:	1 (10)	1 (20)	0 (0)

DONE

CONTRIBUTED PROGRAM **BASIC**

ADDRES
36231

TITLE:

ADDRESS LABELS

DESCRIPTION:

This program allows the printing of addresses on labels. There is an option to update the list of names.

INSTRUCTIONS:

The program is conversational. If you want to stop typing new names, or erasing names, type a zero when the computer asks: NAME?

Two files are needed. The first one will store the data base. The second is used during processing.

You must modify the files statement by typing

```
GET-ADDRES
40  FILES F1,F2
```

You can now run the program.

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

Francois P. Carlhian
Babson College

RUN

```
OPEN-ACOMPF,50
OPEN-DUMMY,50
RUN
ADDRES

**** ADDRES PRINTS ADDRESSES ON LABELS*****



TYPE 0 (ZERO) TO TERMINATE A QUESTION


RESTART-0,NEW ADDRESS-1,PRINT OUT ADDRESSES-2,ERASE ADDRESS-3
?1
TO ENTER NEW NAMES AND ADDRESSES, ANSWER THE QUESTIONS
NAME?JOHN KELLEY
COMPANY?KELLEY ENTERPRISES
ADDRESS?12 BROOK ST.
TOWN?FARM LAND
STATE?WISCONSIN
ZIP CODE?00784
ENTER NEW PERSON
NAME?R. H. SMITH
COMPANY?SMITH INC.
ADDRESS?123 SEAR ST.
TOWN?BELVIEW
STATE?OREGON
ZIP CODE?34567
ENTER NEW PERSON
NAME?BOB MACADAM
COMPANY?A.E. MACADAM INC.
ADDRESS?123 DEARFIELD
TOWN?BAY SHORE,
STATE?NEW YORK
ZIP CODE?11706
ENTER NEW PERSON
NAME?JOHN KELLEY
COMPANY?KELLEY INC.
ADDRESS?12 FARM ST.
TOWN?WARDVILL
STATE?WISCONSIN
ZIP CODE?12345
ENTER NEW PERSON
NAME?BOB SMITH
COMPANY?SMITH INC.
ADDRESS?65 DEAR ST.
TOWN?FREEPORT
STATE?MARYLAND
ZIP CODE?16543
ENTER NEW PERSON
NAME?RICK PEARSON
COMPANY?MARINE SUPPLY
ADDRESS?98 YACHT ST.
TOWN?PORT WASHINGTON
STATE?NEW YORK
ZIP CODE?65743
ENTER NEW PERSON
NAME?0
```

```
RESTART-0,NEW ADDRESS-1,PRINT OUT ADDRESSES-2,ERASE ADDRESS-3
?2
JOHN KELLEY                              R. H. SMITH
KELLEY ENTERPRISES                       SMITH INC.
12 BROOK ST.                             123 SEAR ST.
FARM LAND, WISCONSIN 00784               BELVIEW, OREGON 34567


BOB MACADAM                              JOHN KELLEY
A.E. MACADAM INC.                        KELLEY INC.
123 DEARFIELD                            12 FARM ST.
BAY SHORE,, NEW YORK 11706               WARDVILL, WISCONSIN 12345


BOB SMITH                                RICK PEARSON
SMITH INC.                               MARINE SUPPLY
65 DEAR ST.                              98 YACHT ST.
FREEPORT, MARYLAND 16543                 PORT WASHINGTON, NEW YORK 65743


THERE ARE NO MORE NAMES
RESTART-0,NEW ADDRESS-1,PRINT OUT ADDRESSES-2,ERASE ADDRESS-3
?3
TYPE THE NAME OF THE PERSON YOU WANT TO ERASE?KELLEY
DO YOU WANT TO ERASE JOHN KELLEY KELLEY ENTERPRISES 12 BROOK ST.
FARM LAND WISCONSIN 00784
?NO
DO YOU WANT TO ERASE JOHN KELLEY KELLEY INC. 12 FARM ST. WARDVILL
WISCONSIN 12345
?YES
TYPE THE NAME OF THE PERSON YOU WANT TO ERASE?RICK PEARSON
DO YOU WANT TO ERASE RICK PEARSON MARINE SUPPLY 98 YACHT ST.
PORT WASHINGTON NEW YORK 65743
?YES
TYPE THE NAME OF THE PERSON YOU WANT TO ERASE?SMITH
DO YOU WANT TO ERASE R. H. SMITH SMITH INC. 123 SEAR ST. BELVIEW OREGON
34567
?YES
TYPE THE NAME OF THE PERSON YOU WANT TO ERASE?0
RESTART-0,NEW ADDRESS-1,PRINT OUT ADDRESSES-2,ERASE ADDRESS-3
?1
TO ENTER NEW NAMES AND ADDRESSES, ANSWER THE QUESTIONS
NAME?EDGAR T. CANTY
COMPANY?BB-ABSON COLLEGE
ADDRESS?BABSON PARK
TOWN?WELLESLEY
STATE?MASS.
ZIP CODE?02157
ENTER NEW PERSON
NAME?0
RESTART-0,NEW ADDRESS-1,PRINT OUT ADDRESSES-2,ERASE ADDRESS-3
?2
JOHN KELLEY                              BOB MACADAM
KELLEY ENTERPRISES                       A.E. MACADAM INC.
12 BROOK ST.                             123 DEARFIELD
FARM LAND, WISCONSIN 00784               BAY SHORE,, NEW YORK 11706


BOB SMITH                                EDGAR T. CANTY
SMITH INC.                               BABSON COLLEGE
65 DEAR ST.                              BABSON PARK
FREEPORT, MARYLAND 16543                 WELLESLEY} MASS. 02157


THERE ARE NO MORE NAMES
RESTART-0,NEW ADDRESS-1,PRINT OUT ADDRESSES-2,ERASE ADDRESS-3
?0

DONE
```

ALFTOV
36296

TITLE:

ALPHA TO VARIABLE CONVERSION

DESCRIPTION:

This subroutine will convert a numeric value contained within an alpha string into a number stored in a variable. Blanks are ignored, and any number, positive or negative, integer, decimal, or even written with an "E" exponent will be converted. Conversion stops when a non-valid character is encountered.

INSTRUCTIONS:

The main program must dimension A$, which is the input string. The index of the character in A$ at which the conversion is to start is stored in Z0. The value is returned in the variable Z.

Input Parameters: A$, the string containing the numeric value
Z0, the index of A$ to begin conversion

Output Parameters: Z, the value of the numeric quantity
Z0, the index of the first non-valid character or the end of A$

Entry Point: 9010

Variables Used: Z$(15),(internally dimensioned), Z1, Z2, Z3, Z4, Z5, Z8, and Z9

SPECIAL CONSIDERATIONS:

The length of the subroutine is 433 words. This may be shortened to 282 by removing the unnecessary REM statements.

Overflow or underflow errors will occur if the numeric quantity is outside the limits of the machine.

ACKNOWLEDGEMENTS:

Lawrence E. Turner, Jr.
Pacific Union College

RUN

```
LIS
TEST

10  DIM A$[72]
20  INPUT A$
30  LET Z0=5
40  GOSUB 9010
50  LET X=Z
60  LET Z0=Z0+1
70  GOSUB 9010
80  PRINT "VALUES ARE: ";X;Z
90  GOTO 20

APP-ALFTOV
0-9999 END

RUN
TEST

?AAA=-45.6,78.9006E-7
VALUES ARE: -45.6         7.89006E-06
?AAA=13,-56
VALUES ARE:  13   -56
?AAA=0  7  8  ;-  45E+2
VALUES ARE:  78   -4500
?
DONE
```

CONTRIBUTED PROGRAM **BASIC**

ASCII?
36256

TITLE:

CREATES AN ASCII FILE CONTAINING ALL 256 ASCII CHARACTERS.

DESCRIPTION:

The program "ASCII?" fills the file named "ASCII" with the 256 characters of the ASCII character set. These characters are contained in four 64 character strings, two in record one, and two in record two. The characters are in ASCII order, from lowest to highest.

INSTRUCTIONS:

Open the file "ASCII" two records. The file should have 256 word records. Run the program.

```
GET-ASCII?
OPEN-ASCII,2
RUN
ASCII?

DONE
```

If the file is in the library, it should then be sanctified if possible. The program"?ASCII", (HP36257) creates a list of the characters which are put in the file "ASCII".

SPECIAL CONSIDERATIONS:

~~This~~ program will work only on a 2000~~C system~~. When the file "ASCII" is opened in the library, be sure its records are 256 word records if you plan on using an Aardwolf & Company Writing Team program which uses "ASCII".

The Aardvark and Company Writing Team has designed programs to take up an absolute minimum of computer storage and perform a maximum purpose. The team encourages people to send good programs to Aardvark. As a slight encouragement, the team will give anyone who sends a program which is accepted a free "subscription" to the program handbook and include the contributor as a member of the writing team.

ACKNOWLEDGEMENTS:

Aardvark and Company
2130 Bell Court
Lakewood, Colorado 80215

RUN

```
RUN
ASCII*

DONE
```

CONTRIBUTED PROGRAM **BASIC**

CALNDR
36288

TITLE:

PRINTS A CALENDAR

DESCRIPTION:

This program enables a user to print a calendar for any month or all months for any year.

INSTRUCTIONS:

No special instructions are needed.

The user may answer "Y" or "N" for "Yes/No" questions. A response of "END" to a "Yes" or "No" takes the user to the previous question. A response of "AID" explains any input that is required. A response of "STOP" automatically ends execution of the program.

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

Steve Mudrick
Leasco Response Inc.

RUN

```
RUN
CALNDR


CALENDAR FOR WHAT YEAR?AID
     ENTER A YEAR AFTER 1581 AND BEFORE 8388608.

CALENDAR FOR WHAT YEAR?1973

ANY PARTICULAR MONTH (Y OR N)?N

                    CALENDAR FOR THE YEAR 1973

        JANUARY                  FEBRUARY                  MARCH
 S  M  T  W  T  F  S       S  M  T  W  T  F  S       S  M  T  W  T  F  S
    1  2  3  4  5  6                        1  2  3                  1  2  3
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       4  5  6  7  8  9 10
14 15 16 17 18 19 20      11 12 13 14 15 16 17      11 12 13 14 15 16 17
21 22 23 24 25 26 27      18 19 20 21 22 23 24      18 19 20 21 22 23 24
28 29 30 31               25 26 27 28               25 26 27 28 29 30 31

        APRIL                    MAY                       JUNE
 S  M  T  W  T  F  S       S  M  T  W  T  F  S       S  M  T  W  T  F  S
 1  2  3  4  5  6  7                  1  2  3  4  5                  1  2
 8  9 10 11 12 13 14       6  7  8  9 10 11 12       3  4  5  6  7  8  9
15 16 17 18 19 20 21      13 14 15 16 17 18 19      10 11 12 13 14 15 16
22 23 24 25 26 27 28      20 21 22 23 24 25 26      17 18 19 20 21 22 23
29 30                     27 28 29 30 31            24 25 26 27 28 29 30

        JULY                     AUGUST                    SEPTEMBER
 S  M  T  W  T  F  S       S  M  T  W  T  F  S       S  M  T  W  T  F  S
 1  2  3  4  5  6  7                  1  2  3  4                        1
 8  9 10 11 12 13 14       5  6  7  8  9 10 11       2  3  4  5  6  7  8
15 16 17 18 19 20 21      12 13 14 15 16 17 18       9 10 11 12 13 14 15
22 23 24 25 26 27 28      19 20 21 22 23 24 25      16 17 18 19 20 21 22
29 30 31                  26 27 28 29 30 31         23 24 25 26 27 28 29
                                                    30

        OCTOBER                  NOVEMBER                  DECEMBER
 S  M  T  W  T  F  S       S  M  T  W  T  F  S       S  M  T  W  T  F  S
    1  2  3  4  5  6                        1  2  3                        1
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       2  3  4  5  6  7  8
14 15 16 17 18 19 20      11 12 13 14 15 16 17       9 10 11 12 13 14 15
21 22 23 24 25 26 27      18 19 20 21 22 23 24      16 17 18 19 20 21 22
28 29 30 31               25 26 27 28 29 30         23 24 25 26 27 28 29
                                                    30 31
```

```
MORE (Y OR N)?Y

CALENDAR FOR WHAT YEAR?2000

ANY PARTICULAR MONTH (Y OR N)?Y

WHAT MONTH?2
```

FEBRUARY 2000

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29				

```
MORE (Y OR N)?N

DONE
```

CONTRIBUTED PROGRAM **BASIC**

CHARS
36220

TITLE: ASCII CHARACTER SET

DESCRIPTION:

This file (A000,CHARS) contains the complete ASCII character set. It is composed of two 64-word blocks; each block contains a 64-character string.

INSTRUCTIONS:

1. Declare the file $CHARS in your BASIC program (e.g., '10 FILES $CHARS').
2. Read one, or both, of the strings into a 64-character string array (e.g., '50 READ #1; A$,B$').
3. Extract those ASCII characters required for the application from the string(s) (e.g., '100 Z$(1,5) = B$(12,16)').
4. Utilize the selected ASCII characters as required in the application (e.g., '800 PRINT Z$(3,3), I,J,K,Z$(1,1)').

SPECIAL CONSIDERATIONS:

Distribution of 'A000,CHARS' is available via magnetic tape only. The characters are not all conducive to paper tape I/O. Control characters for the HP 2600 are included in the file. See ASCII Chart on Page 2.

ACKNOWLEDGEMENTS:

Ralph Carpenter
Hewlett-Packard/Data Systems

ASCII Character Set

		CONTROL CHARACTERS		GRAPHIC CHARACTERS					
	COL.	0	1	2	3	4	5	6	7
ROW	BITS	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P		p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
10	1010	LF	SUB	*	:	J	Z	j	z
11	1011	VT	ESC	+	;	K	[	k	{
12	1100	FF	FS	,	<	L	\	l	|
13	1101	CR	GS	-	=	M	]	m	}
14	1110	SO	RS	.	>	N	^	n	~
15	1111	SI	US	/	?	O	_	o	DEL

CHARSE
36757

TITLE:

ASCII CHARACTER SET FOR 2000E

DESCRIPTION:

CHARSE is a file which contains all 128 ASCII characters. The file is organized as two character strings, each 64 long.

INSTRUCTIONS:

To use, add the following to your program:

```
10  FILES  CHARSE
20  DIM X$(64),Y$(64)
     .
     .
     .
     .
30  READ #1;X$,Y$
     .
     .
     .
     .
```

Then X$ and Y$ would contain the characters.

continued on following page

SYSTEM SPECIFICATIONS:

2000E (ONLY) and Teletype

SPECIAL CONSIDERATIONS:

This file is created for the express purpose of printing characters which BASIC will not allow as input. The user is cautioned against using these strings in comparisions, etc. In order to work properly, the characters are all stored with their parity bit on. Inputted characters all have their parity bit off. In practice, this means an "A" from CHARSE will not match an "A" which is typed-in. (The parity bits are all stripped off by the BASIC output routine and the hardware then generates even parity.)

See HP 36220 (CHARS) for an ASCII file for the HP 2000C/F.

ACKNOWLEDGEMENTS:

Ted Park
Pacific Union College

INSTRUCTIONS continued

ASCII CHARACTER FILE "CHARS" STRING 1

INDEX	ASCII	OCTAL	INDEX	ASCII	OCTAL
1	NULL	00	33	SPACE	40
2	SOH	01	34	!	41
3	STX	02	35	"	42
4	ETX	03	36	#	43
5	EOT	04	37	$	44
6	ENQ	05	38	%	45
7	ACK	06	39	&	46
8	BELL	07	40	'	47
9	BS	10	41	(	50
10	HT	11	42	)	51
11	LF	12	43	*	52
12	VT	13	44	+	53
13	FF	14	45	,	54
14	CR	15	46	-	55
15	SO	16	47	.	56
16	SI	17	48	/	57
17	DLE	20	49	0	60
18	DC1	21	50	1	61
19	DC2	22	51	2	62
20	DC3	23	52	3	63
21	DC4	24	53	4	64
22	NAK	25	54	5	65
23	SYN	26	55	6	66
24	ETB	27	56	7	67
25	CAN	30	57	8	70
26	EM	31	58	9	71
27	SUB	32	59	:	72
28	ESC	33	60	;	73
29	FS	34	61	<	74
30	GS	35	62	=	75
31	RS	36	63	>	76
32	US	37	64	?	77

ASCII CHARACTER FILE "CHARS" STRING 2

INDEX	ASCII	OCTAL	INDEX	ASCII	OCTAL
1	@	100	33	`	140
2	A	101	34	a	141
3	B	102	35	b	142
4	C	103	36	c	143
5	D	104	37	d	144
6	E	105	38	e	145
7	F	106	39	f	146
8	G	107	40	g	147
9	H	110	41	h	150
10	I	111	42	i	151
11	J	112	43	j	152
12	K	113	44	k	153
13	L	114	45	l	154
14	M	115	46	m	155
15	N	116	47	n	156
16	O	117	48	o	157
17	P	120	49	p	160
18	Q	121	50	q	161
19	R	122	51	r	162
20	S	123	52	s	163
21	T	124	53	t	164
22	U	125	54	u	165
23	V	126	55	v	166
24	W	127	56	w	167
25	X	130	57	x	170
26	Y	131	58	y	171
27	Z	132	59	z	172
28	[	133	60	{	173
29	\	134	61	|	174
30	]	135	62	}	175
31	↑	136	63	~	176
32	←	137	64	DEL	177

TITLE: DATE AND DAY OF THE WEEK

DATER
36298

DESCRIPTION:

This is a subroutine to return the current date and day of the week in two alpha strings.

INSTRUCTIONS:

The current date is returned as: dd MON yy
where

dd	is an integer day of the month
MON	is a three letter abbreviation of the month
yy	is the last two digits of the year

The day of the week is returned as a three letter abbreviation.

Output Parameters: N$, the current date (internally dimensioned: N$(9))
M$, the current day of the week (internally dimensioned: M$(36), logical length is 3)

Entry Point: 9810

Variables Used: M$(36), n$(9), Z(12), Z$(10), Z1, Z2, Z3, and Z8

SPECIAL CONSIDERATIONS:

The length of the subroutine is 536 words. This may be shortened to 392 words by the deletion of the REM statements.

ACKNOWLEDGEMENTS:

Lawrence E. Turner, Jr.
Pacific Union College

```
RUN

LIS
TEST

10  GOSUB 9810
20  PRINT N$,M$
30  STOP

APP-DATER
9999 END

RUN
TEST

27 MAR 73       TUE

DONE
```

CONTRIBUTED PROGRAM **BASIC**

EDIT2K
36838

TITLE:

TEXT EDITOR FOR THE HP 2000C AND 2000C'/F

DESCRIPTION:

EDIT2K is a contributed utility for use on an HP 2000C, or HP 2000C High-Speed, or HP 2000F, Time-Shared BASIC (TSB) Operating System.

An EDIT2K user needs only minimal experience with typewriting to write a text, store it in a TSB file, and make changes or corrections to the text. Once such a text file is created it can be recalled at any time to produce copies of the text, or to be updated, or to be edited and copied into another file as the basis for another text.

To accomplish its functions, EDIT2K accepts a wide variety of commands from a user. Most of the commands use the same expressions one might use to give instructions to a typist or secretary. Others use special notations required by EDIT2K. In all cases, use of the command is easily learned by any typewriter user.

EDIT2K is contributed in five programs: EDIT2K, EDITA, EDITB, EDITC, and EDITD. Each is to be LOADed into a TSB system (public) library (i.e., under IDcode A000). Any one can be used to initiate EDIT2K operation.

INSTRUCTIONS:

See next page.

SPECIAL CONSIDERATIONS:

This documentation was produced using the EDIT2K program.

ACKNOWLEDGEMENTS:

Donald R. Coleman
HP/Data Systems Division

INSTRUCTIONS:

OPERATION

Operation of EDIT2K includes four general activities: Logging on to the TSB, opening files, calling EDIT2K, and typing EDIT2K commands.

LOGGING ON TO THE TSB

To log on to the TSB a user must have an IDcode and password for the the TSB to be used. Then the IDcode and password must be typed on a keyboard terminal connected to that TSB. All EDIT2K commands will be typed on the same keyboard terminal used to log on.

For example, type

HELLO-C913,GNOMON

The TSB responds with a message to welcome the user to the system. See the manual for the TSB to be used for full details.

OPENING FILES

Before EDIT2K can be called from the TSB system library, at least two files must be opened for use by EDIT2K: one for a scratch or working file and one for a keep or permanent text file. A hold or temporary text file can be opened too, but is not required.

The scratch file is the one in which all EDIT2K work is done. That is, it is where a new text is written or an old text is changed or modified.

The keep file is one where a permanent copy of the text is saved for any future use, including return to the scratch and/or hold files for further changes.

The hold file is one where part of a scratch file can be temporarily saved, while other work is done in the scratch file, then returned to the scratch file to complete a text. Or, text from the scratch file can be appended to the text in the hold file.

More than one file can be opened for any of the three types of EDIT2K files. For example, one could open a scratch file, a keep file, and a hold file for each text to be written. Or, one scratch file and one hold file might be used in common for several keep files. The latter method conserves file space on a TSB and is therefore recommended.

File Size Requirements

Each type of file required by EDIT2K should meet the following size minimums to contain one page (66 lines) of text:

Scratch files need 14 to 70 records (sectors) of 256 words each, 14 to create the text and up to 56 more for editing operations.

Keep files need 1 to 11 records (sectors) of 256 words each, 1 for one character per line or 11 for 72 characters per line.

Hold files need the same size as keep files.

INSTRUCTIONS: (Continued)

How To Open Files

Full instructions are given in the manual for the TSB to be used. Briefly, they state: Type an OPEN command for each file to be opened, in this format:

```
OPEN-SCR,500
```

where SCR is a sample file name and 500 is a sample file size.

CALLING EDIT2K

After at least a scratch and a keep file have been opened, EDIT2K can be called into operation from the TSB system (public) library. Full instructions are given in the manual for the TSB to be used. Briefly, they state: Type

```
GET-$EDIT2K



RUN
```

EDIT2K execution starts with a series of four questions:

EDIT FILE? Type the name of the scratch file to be used.

HOLD FILE? Type the name of the hold file to be used, or, if none is to be used, strike the RETURN (or equivalent) key.

RECOVERY? The normal response is NO. However, if EDIT2K is being restarted after a previous run was interrupted (perhaps by the BREAK key), the response is YES.

ARE YOU RE-ENTERING A 'KQ' FILE? The normal response is NO. However, if EDIT2K is being restarted after a previous run in which the command KEEPQ (or KQ) was used to save a text in the scratch file, the response is YES.

TYPING EDIT2K COMMANDS

Whenever it is ready to accept commands, EDIT2K prints a slash / as the prompt character. Type any EDIT2K command, using the following instructions.

TSB Special Control Keys

If a wrong character is used while typing a line, use the character left-arrow <- to erase that wrong character. Or use the left-arrow several times to erase several successive wrong characters.

To correct an entire line before that line is terminated by the RETURN (or equivalent) key, use CONTROL-X (press and hold the CONTROL key and strike X).

EDIT2K Special Characters

EDIT2K uses five special characters:

The slash is printed by EDIT2K to request (prompt) an EDIT2K command.

The caret is printed by EDIT2K in response to a FIND command to show the line found and in which the next EDIT2K operation would begin. For example, FIND might be used to locate a line that contains a given character or phrase. Then the MODIFY command could be typed without a parameter to make a change in that current line.

INSTRUCTIONS: (Continued)

* The asterisk can be typed as a parameter in an EDIT2K command to specify the current position of the EDIT2K pointer (as shown by the caret ^).

; The semicolon can be typed in an EDIT2K command line to include more than one command in that line. The semicolon separates each command from the preceeding command.

CONTROL-Y
The CONTROL-Y (press and hold the CONTROL key and strike Y) is typed to terminate text entries to any EDIT2K operation.

NOTE: Alphabetic characters in any EDIT2K command name or special character must be typed in upper-case. Thus, if both upper- and lower-case are being written into a text, to type a command name or a special character such as CONTROL-Y the shift key must be used too.

Line Numbering

Line numbers in an EDIT2K text use the form

iiii.dd

where iiii are integer digits (from 1 to 9999) and dd are decimal digits (from 00 to 99). When a new text is written using the ADD command, line numbers are all integers starting with 1 and incrementing by 1. This allows for later insertions of new lines between original lines.

The increment for new lines can be changed by using the SET command.

When specifying line numbers leading integer zeroes and trailing decimal zeroes can be omitted.

When the KEEP command is used to save a text in a keep file, an UNNUMBERED option can be specified. If so, the original line numbers are not written into the keep file. Subsequently when that keep file is recalled by the TEXT command, new line numbers will be assigned starting with 1 and incrementing by 1.

Parameters

Many EDIT2K commands use one or more parameters to name parts of a text to be processed by the command. For example, one or more lines or one or more characters can be named. A guide to parameters begins on the next page, for use in the EDIT2K command format definitions on the pages that follow.

INSTRUCTIONS: (Continued)

Parameter
Syntax
Examples and Descriptions, where needed.

<range>
ALL¦<position>¦<position>/<position>¦<null>

112	Line number 112, left-most column.
35.02	Line number 35.02, left-most column.
63+4	The fourth line after line 63.
213-2	The second line before line 213.
476(15)	Line 476, fifteenth column from the left.
132(+4)	Line 132, the fourth non-blank character.
65(-3)	The third non-blank character before line 65 (from the end of the preceeding line).
32/65	All of lines 32 through 65.
301/401,604/689	Lines 301 through 401 and lines 604 through 689, but not 401.01 through 603.99.
222(13)/234(36)	Line 222 column 13 through line 234 column 36.
"GNOMON"	The next occurence of the string GNOMON after the current column position of the pointer.
181/"Boot"	Line 181 through the character t in the next appearance of the characters Boot.
*	The current line or column position of the pointer.
*+3	The third line after the current line.
*-6	The sixth line before the current line.
*(+2)	The second non-blank character after the current column position of the pointer (may be in the next line).
*(-4)	The fourth non-blank character before the current column position of the pointer (may be in the preceeding line).
/(LAST)	From the current column position of the pointer through the last column in the same line.
/+3	From the current column position of the pointer through the third line after the current line.
/(+5)	From the current column position of the pointer to the fifth non-blank character after that current column (may be in the next line).

<range list>
<range>¦<range>,<range list>

<position>
<record position>¦<record position>(<column position>)

<record position>
<record identifier>¦<record identifier><relative record position>

<record identifier>
<line number>¦<string>¦*¦FIRST¦LAST

<line number>
<unsigned integer>¦<unsigned fraction>¦<unsigned real number>

<unsigned integer>
1¦2¦3¦ . . . ¦9997¦9998¦9999

<unsigned fraction>
.00¦.01¦.02¦ . . . ¦.97¦.98¦.99

<unsigned real number>
<unsigned integer><unsigned fraction>

<string>
Any ASCII graphic characters delimited by any special character other than , (comma) ; (semicolon) . (period) ((left parenthesis)) (right parenthesis) * (asterisk) / (slash) + (plus sign) - (minus sign).

INSTRUCTIONS: (Continued)

Parameter
Syntax
Examples and Descriptions, where needed.

<relative record position>
<signed integer>

<signed integer>
+<unsigned integer>¦-<unsigned integer>

<column position>
<relative character position>¦<absolute column position>

<relative character position>
<signed integer>

<absolute column position>
<column specifier>¦<column specifier><absolute column adjustment>

<column specifier>
<unsigned integer>¦LEFT¦RIGHT¦LAST¦*

<absolute column adjustment>
<signed integer>

<increment>
<unsigned integer>¦<unsigned fraction>

<line length>
1¦2¦3¦ . . . ¦70¦71¦72

<set option list>
<set option>¦<set option>,<set option list>

<set option>
FROM¦DELTA¦LEFT¦RIGHT¦LENGTH¦QUIET¦DISPLAY

EDIT2K Command Formats

In each command, the full name can be shortened to only the first character (i.e., ADD can be A, REPLACE can be R, etc.). See the parameter expessions definitions in the preceeding pages. Use CONTROL-Y to end use of a command unless other stated.

ADD[Q] [<line number>][,HOLD[Q][,NOW]]
To add lines of text to the scratch file.

If the Q option is not used, ADD prints the line number to be added then waits for typed inputs to the text.

If the ,HOLD or ,HOLDQ or ,HOLD,NOW or ,HOLDQ,NOW options are used, ADD copies from the hold file into the scratch file. The Q option for HOLD omits printing of the lines copied from the hold file. Without the ,NOW option, CONTROL-Y must be used before the ADD from the hold file begins.

CHANGE[Q] <absolute column position>[/<absolute column position>]
TO<string>[IN<range list>]
-or-
CHANGE[Q] <string>TO<string>[IN<range list>]
-or-
CHANGE[Q] ""TO<string>[IN<range list>]
To change text in the scratch file.

The words TO and FROM can be replaced by commas. If the Q option is used, the new line content is not printed. If only one <absolute column position> is used, or if the third format shown is used, the TO<string is used only once, as an insert before the <range list> or the current position of the pointer.

INSTRUCTIONS: (Continued)

DELETE <range list>
To erase lines of text from the scratch file.

END
To terminate EDIT2K operation. Do not use END before the KEEP command has been used to save the scratch file contents.

FIND[Q] <range>
To find text in the scratch file.

If the Q option is not used, FIND lists the contents of the the line found.

GATHER[Q] <range>TO<line number>[BY<increment>]
To move text to another location in the scratch file.

The words TO and BY can be replaced by commas. If the Q option is not used, GATHER reports the old line numbers versus the new line numbers.

HOLD[Q] <range>[,APPEND]
To copy text from the scratch file into a hold file.

If the Q option is not used, HOLD lists the entire text copied into the hold file. If the ,APPEND option is used, the scratch file contents in the <range> will be appended onto the current end of the hold file. To clear the hold file, use HOLD without parameters.

INSERT[Q] <position>[BY<increment>][,HOLD[Q][,NOW]]
To insert new text into the scratch file.

If the Q option is used, INSERT will not print the line in which the insertion is to be made, but it will prompt inputs by ringing the bell (if the terminal has one). The BY<increment> option can can be used to temporarily use a line numbering increment other than that in use (from the default or from the SET command).

If the ,HOLD or ,HOLDQ or ,HOLD,NOW or ,HOLDQ,NOW options are used, INSERT copies from the hold file into the scratch file. The Q option for HOLD omits printing of the lines copied from the hold file. Without the ,NOW option, CONTROL-Y must be used before the INSERT from the hold file begins.

JOIN[Q] file name[(<line number>/<line number>)][TO<line number>] [BY<increment>]
-or-
JOIN[Q] file name[(#<unsigned integer>/#<unsigned integer>)] [TO<line number>][BY<increment>]
To add text from a keep file to text in the scratch file.

JOIN is similar to GATHER but for these exceptions:

1. It copies lines from another keep file, rather than from other parts of the scratch file.
2. The optional (<line number>/<line number>) refers to line numbers in that keep file to copied. Or, the optional (#<unsigned integer>/#<unsigned integer>) refers to the sequence in which the lines in the other keep file appear.

INSTRUCTIONS: (Continued)

KEEP[Q] file name[(<range>)][,UNNUMBERED]
To copy text from the scratch file into a keep file.

If the Q option is used, the scratch file becomes a special keep file in the edit format. To recall that file, it must be named the scratch file as before then YES must answered to the question ARE YOU RE-ENTERING A 'KQ' FILE? (see "CALLING EDIT2K"). Do not specify a file name or a <range> or UNNUMBERED when the KEEPQ form is used.

If the UNNUMBERED option is used, the original line numbers are not written into the keep file. Subsequently, when that keep file is recalled by the TEXT command, the UNNUMBERED option for the TEXT command must be used. New line numbers will be assigned starting with 1 and incrementing by 1.

LIST[Q] <range>
To list text from the scratch file.

If the Q option is used, no line numbers will be listed.

If a <range> is not used, only the line currently pointed to will be printed.

If ,NOTEXT is used, no text will be printed. If both Q and ,NOTEXT are used, the ,NOTEXT option is ignored.

MODIFY[Q] <range list>
To change text in the scratch file.

If the Q option is used, the new, edited version of the line is not printed.

MODIFY prints, one at a time, each line in the range list to allow use of a D (delete), I (insert), and/or R (replace) sub-command for each line, as follows:

To delete three characters in line 123, use the space bar (or the space bar and the REPEAT key) to move the carriage under first character to be deleted then type D D as follows:

```
123    Now is the day for all good men.
                          D D
123    Now is the day for good men.
```

If the new version is correct, strike the RETURN (or equivalent) key; if not, use the D, I, or R sub-command accordingly or use CONTROL-Y to keep the original version.

To insert a new word in line 41, move the carriage (as before) under the character before which the insertion is to be made, then type Icharacters as follows:

```
41    Today is the day of the week.
                    Ifourth
41    Today is the fourth day of the week.
```

The characters fourth include a blank space that does not show in the above example but it must be typed to be inserted. As before, if the new version is correct, strike the RETURN key; if not, use D, I, R, or CONTROL-Y.

To replace characters in line 1113, move the carriage (as before) under the first character to be replaced then type Rcharacters as follows:

```
1113    here are sample problems.
                   Rtwelve
1113    here are twelve problems.
```

As before, if the new version is correct, strike the RETURN key; if not, use D, I, R, or CONTROL-Y.

INSTRUCTIONS: (Continued)

Three special notes:
1. To insert new characters at the end of a line, type the I subcommand after the last character in the existing line.
2. The second D for the D sub-command can be replaced by Icharacters to insert new characters in place of those characters deleted.
3. To discontinue a current use of MODIFY, use CONTROL-Y; the original versions of each line in the range list> will remain intact.

Q <string>
To be used only in a keep file for the USE command.

Q as command by itself is used only in file written for the USE command, to print the <string> on the keyboard terminal (or on the line printer if one is in use).

REPLACE[Q] <range list>[,HOLD[Q][,NOW]]
To replace lines of text in the scratch file.

REPLACE is similar to ADD but for these exceptions:
1. If the Q option is used, the previous contents of the lines in the <range list> will not be printed (otherwise they are, one at a time).
2. The line number to be replaced is always printed before new text will be accepted.

SET [[,]FROM=<line number>][[,]DELTA=<increment>][[,]LEFT=<column position>][[,]RIGHT=<column position>][[,]LENGTH=<line length>] [[,]QUIET!DISPLAY]
To change EDIT2K operating parameters.

When EDIT2K operation starts, each parameter is set to a default value by the equivalent of this SET command:
SET FROM=1,DELTA=1,LEFT=1,RIGHT=72,LENGTH=72,DISPLAY

SET enables each parameter to be changed at any time during use of EDIT2K. To do so, write the SET command and each <set option> wanted in any order, but follow each <set option> except the last with a comma.

FROM= is used to define the first line number used if the TEXT command is used with its ,UNNUMBERED option.

DELTA= defines the increment to be used between new line numbers.

LEFT= and RIGHT= define the first and last columns in a line that EDIT2K will process through any command except GATHER and JOIN and KEEP which ignore the LEFT and RIGHT limits.

LENGTH= defines the maximum line length to which a line can be increased by any EDIT2K command.

TEXT file name[(<line number>/<line number>)][,UNNUMBERED]
-or-
TEXT file name[(#<unsigned integer>/#<unsigned integer>)] [,UNNUMBERED]
To copy a keep file into the scratch file.

TEXT copies the keep file named file name into the scratch file for EDIT2K operations. Any current contents of the scratch file are lost, be sure to consider using the KEEP command before using this TEXT command.

The optional (<line number>/<line number>) specifies actual line numbers in the keep file named that will be copied.

The optional (#<unsigned integer>/#<unsigned integer>) specifies the sequence in which the line in the keep file named appear.

INSTRUCTIONS: (Continued)

The optional ,UNNUMBERED directs TEXT to ignore the line numbers used in the keep file named and assign new numbers starting with 1 and incrementing by 1 (unless SET has specified otherwise).

USE file name
To read EDIT2K commands from a keep file.

USE directs EDIT2K to read commands from a keep file named <file name> rather than from the keyboard terminal. Such a USE file can be written through EDIT2K normal operations.

VERIFY ALL or VERIFY <set option list>
To report the current <set option list> in effect from either the EDIT2K default values or from the last use of the SET command.

Table 1. EDIT2K Error Messages

1. INVALID COMMAND NAME
2. INVALID OPTION
3. INVALID OR MISSING PARAMETER
4. INVALID RANGE
5. INVALID COLUMN RANGE
6. ABSOLUTE COLUMN POSITION OUT OF RANGE
7. INVALID LINE NUMBER
8. INVALID INTEGER
9. INVALID STRING DELIMITER
10. UNDELIMITED STRING
11. FAILURE TO OPEN FILE
12. WHILE FILE OVERFLOW
13. FILE NOT ACCESSABLE (x)
14. FILE STRUCTURE FULL-'KEEP' AND 'TEXT' TO RESTPUCTURE
15. AMBIGUOUS REQUEST - THE FILE IS UNNUMBERED
16. POSITION NOT FOUND
17. STRING NOT FOUND
18. LINE DOES NOT EXIST
19. COMMAND WILL NOT REPLACE OR INTERLEAVE LINES
20. LINE ZERO CANNOT BE ACCESSED
21. END-OF-FILE

CONTRIBUTED PROGRAM **BASIC**

TITLE:

FILE MANIPULATION - CREATES, EDITS, LISTS, SORTS, EMULATES G.E. MK II.

EDITOR
36749

DESCRIPTION:

These programs attempt to facilitate working with an internal file on an HP Time Sharing system. The overall philosophy is that a person with the barest minimum knowledge should be able to add and delete records. Later he could then learn to use the various options as the need arises. Adding a record to a file is exactly like adding a line to a BASIC program. Deleting a line is also the same. In both cases line numbers are used to address program statements. The editor's primary addressing is also by line numbers. Internally a line number is mapped as a logical record onto a physical record displaced by an offset. This means that if 10 logical records per physical record are defined, with 50 physical records; 500 line numbers are available for data. (In this case each line could only have 50 characters because there are 512 characters per physical record). The user need only be concerned with line numbers. These line numbers may be renumbered to allow inserting data when the lines become contiguous. Also, an interval value like 1.1 will cause every 10th line to be skipped, which is nice for a full file requiring programmed inserts, (when the renumber command is used).

The LIST command lists all active (non-null) records and their line numbers, starting at line number F and going to line number L (prompt is F, L?). These are the very basic tools and allow a person to control the content of a file much as he would control the content of a BASIC program.

INSTRUCTIONS:

A file by the name of DEL is required. This file is used to buffer records and needs to be 3 records long. Your working file must be opened, and in addition, a directory file must be opened. The directory file must be named the same name as your working file, followed by a "D"- e.g., WORK and WORKD. The directory file must be 20 blocks long.

HELP COMMAND

HOW IT WORKS: ENTER 'HELP' TO EDITOR PROMPT. Gives a list of commands and instructions for operating the EDITOR program.

(Continued on next page)

ACKNOWLEDGEMENTS:

Denis Ferland
HP/Midwest Sales Region

INSTRUCTIONS: (Continued)

FILE COMMAND

HOW IT WORKS: ENTER 'FILE' TO EDITOR PROMPT. Allows the editor to change the file it is operating on.

1. Enter next file to be operated on in response to 'FILE NAME'.

2. If file does not exist an error message will cause the program to abort.

3. If a new file is called it is automatically initialized by the Editor.

   Physical records: Answer # of records used in open command. Number of logicals per physical: Answer # of records to be contained in 512 characters available in one physical record.

FIND COMMAND

HOW IT WORKS: ENTER 'FIND' TO EDITOR PROMPT. A particular record is located (or determined to be missing by means of a key given by an input String). The length of the key is the same length as the input String. A combination linear hash and Table correction is used to arrive at the correct record number in the minimum number of file accesses.

CHANGE COMMAND

HOW IT WORKS: ENTER 'CHANGE' TO EDITOR PROMPT. Will search for occurrences of 'FROM' string and replace them with occurrences of 'TO' string.

Operation Options:

1. The answer 'YES' to the question 'verify?' will allow the user to over-ride each change before it is executed.

2. 'F, L?' is the request for line numbers to be effected by 'CHANGE'.

POSTFIX COMMAND

HOW IT WORKS: ENTER 'POST FIX' TO EDITOR PROMPT. Will add a given string to the end of a record.

Operation:

1. 'STRING' is information to be added to end of record.

2. 'F, L' is the request for line numbers to be effected by 'POSTFIX'.

PREFIX COMMAND

HOW IT WORKS: ENTER 'PRE FIX' TO EDITOR PROMPT. Will add a given string to front of a record.

Operation:

1. 'STRING' is information to be added to the front of record.

2. 'F, L' is the request for line numbers to be effected by 'PREFIX'.

LOCATE COMMAND

HOW IT WORKS: ENTER 'LOCATE' TO EDITOR PROMPT. Will sequentially scan records until an occurrence of a given String is found. It will print the record in which the string occurs at least once.

Operation:

1. 'STRING' is the string to be found.

2. 'F, L' requests the line numbers to be effected by 'LOCATE'.

DELETE CHARACTER COMMAND

HOW IT WORKS: ENTER 'DELETE CHARACTER' TO EDITOR PROMPT. Will delete a given sequence of characters with a record.

Operation:

1. From character number, to character number requests definition of the interval of characters to be deleted.

2. 'F, L' is the request for line numbers to be effected by 'DELETE'.

INSTRUCTIONS: (Continued)

INSERT LINE

HOW IT WORKS: ENTER 'INSERT LINE' TO EDITOR PROMPT. A new record is inserted after the given line number. The contiguous line numbers are adjusted accordingly.

INSERT CHARACTER COMMAND

HOW IT WORKS: ENTER 'INSERT' TO EDITOR PROMPT. Will insert a given string after the given character number.

Operation:

1. 'AFTER' position is the request for insert position.

2. 'F, L' is the request for line numbers to be effected by 'INSERT'.

NOTE: A record that is too short will not be affected.

RENUMBER COMMAND

HOW IT WORKS: ENTER 'RENUMBER' TO EDITOR PROMPT. Allows the line numbers to be resequenced by a specified interval. Fractional intervals such as '.1' will cause every 10th line to be blank.

A check is made to determine if file is large enough to accomodate a particular interval. Number of active and inactive records is printed.

LIST SORTED COMMAND

HOW IT WORKS: ENTER 'LIST SORTED' TO EDITOR PROMPT. Sorting is automatic and continous as file is built. The 'LIST SORTED' command generates a list of the file in a sorted manner.

DONE; END; STOP; BYE

These commands will cause termination of the editor program.

```
RUN

OPE-DEL,3
OPE-WORK,20
OPE-WORKD,20
GET-EDITOR
RUN
EDITOR

FILENAME?WORK
MAX# OF PHYSICAL RECORDS?20
NUMBER OF LOGICALS PER PHYSICAL?7
MAX STRING SIZE IS-72
?HELP
LEGAL COMMANDS ARE: LIS,LIST,LIST SORTED
                    FILE (TO CHANGE THE FILE YOU ARE WORKING ON
                      DELETE LINES, DELETE CHARACTERS
                      INSERT LINES , INSERT CHARACTERS
                      LOCATE (KEYWORD SEARCH)
                      FIND (FAST LOOKUP)
                     POSTFIX,PREFIX
                     RENUMBER
                     GENERATE TEST FILE
                     HELP
                     END,STOP,DONE,BYE
ALL LINES ARE ADDED AND DELETED BY MEANS OF LINE NUMBERS
JUST LIKE IN WRITING A BASIC PROGRAM
A DIRECTORY FILE MUST BE OPENED 20 BLOCKS LONG HAVING THE
 THE SAME NAME AS THE FILE YOU WISH TO CREATE
FOLLOWED BY A 'D'. A FILE CALLED 'DEL' IS NEEDED FOR SCRATCH SPACE
IT NEEDS JUST THREE BLOCKS
THEREFORE TO USE THE FILE  'WORK'  :
OPEN-WORK,50
OPEN-WORKD,20
OPEN-DEL,3
THAT SHOULD PUT YOU IN BUSINESS!!!!
F,L?  MEANS FIRST TO LAST LINE NUMBERS TO BE AFFECTED
?1  THIS IS A STATEMENT LINE TO BE ADDED TO WORK FILE
```

```
?2  PUMPKIN
?3  NEGATIVE
?6  FLOWERS
?8  GRAPES
?15  LAWRENCE
?20  ELEPHANTS
?25  ZEROX
?23  TOMATOES
?LIST SORTED
 20   ELEPHANTS
 6    FLOWERS
 8    GRAPES
 15   LAWRENCE
 3    NEGATIVE
 2    PUMPKIN
 1    THIS IS A STATEMENT LINE TO BE ADDED TO WORK FILE
 23   TOMATOES
 25   ZEROX
LIST SORTED DONE
?CHANGE
VERIFY?YES
CHANGE FROM:?P
          TO:?G
F,L
?1,25
OLD 2    PUMPKIN
NEW 2    GUMPKIN
CHANGE?YES
OLD 2    GUMPKIN
NEW 2    GUMGKIN
CHANGE?YES
OLD 8    GRAPES
NEW 8    GRAGES
CHANGE?YES
OLD 20   ELEPHANTS
NEW 20   ELEGHANTS
CHANGE?YES
CHANGE DONE
?CHANGE
VERIFY?YES
CHANGE FROM:?G
          TO:?P
F,L
?1,25
OLD 2    GUMGKIN
NEW 2    PUMGKIN
CHANGE?YES
OLD 2    PUMGKIN
NEW 2    PUMPKIN
CHANGE?YES
OLD 3    NEGATIVE
NEW 3    NEPATIVE
CHANGE?NO
OLD 8    GRAGES
NEW 8    PRAGES
CHANGE?NO
OLD 8    GRAGES
NEW 8    GRAPES
CHANGE?YES
OLD 20   ELEGHANTS
NEW 20   ELEPHANTS
CHANGE?YES
CHANGE DONE
?LIS
F,L?1,25
 1    THIS IS A STATEMENT LINE TO BE ADDED TO WORK FILE
 2    PUMPKIN
 3    NEGATIVE
 6    FLOWERS
 8    GRAPES
 15   LAWRENCE
 20   ELEPHANTS
 23   TOMATOES
 25   ZEROX
LIST DONE
?LIST SORTED
 20   ELEPHANTS
 6    FLOWERS
```

```
 8     GRAPES
 15    LAWRENCE
 3     NEGATIVE
 2     PUMPKIN
 1     THIS IS A STATEMENT LINE TO BE ADDED TO WORK FILE
 23    TOMATOES
 25    ZEROX
LIST SORTED DONE
?FIND
KEY?T
 1     THIS IS A STATEMENT LINE TO BE ADDED TO WORK FILE
 23    TOMATOES
KEY?15
KEY?L
 15    LAWRENCE
KEY?G
 8     GRAPES
KEY?A
KEY?B
KEY?C
KEY?D
KEY?E
 20    ELEPHANTS
KEY?F
 6     FLOWERS
KEY?G
 8     GRAPES
KEY?H
KEY?I
KEY?J
KEY?K
KEY?L
 15    LAWRENCE
KEY?M
KEY?N
 3     NEGATIVE
KEY?O
KEY?P
 2     PUMPKIN
KEY?Q
KEY?R
KEY?S
KEY?T
 1     THIS IS A STATEMENT LINE TO BE ADDED TO WORK FILE
 23    TOMATOES
KEY?ENE←D
EXITED FROM FIND MODE
?POSTFIX
STRING?OF ARABIA
F,L?15,15
DONE
?PREFIX
STRING?HALLOWEEN
F,L?2,2
DONE
?LOCATE
STRING?LAWRENCE
F,L?1,10←8
 15             LAWRENCEOF ARABIA
DONE
```

CONTRIBUTED PROGRAM **BASIC**

FGRAPH
36165

TITLE: SIMULTANEOUS FUNCTION GRAPHER

DESCRIPTION:

This program graphs from one to five functions simultaneously in an interval between two points, providing, for each value in that interval, the calculation of the functions does not involve square roots or negative numbers, overflows, or other error conditions.

The program adjusts the scale of the Y-axis so that all values of the function in the interval appear on the graph, and places the X-axis accordingly.

INSTRUCTIONS:

Type in the functions to be graphed starting the FNA at line 10 and continuing with FNB at line 20 to a maximum of five. Type RUN. The program will ask for the last letters of the functions FN(_) to be graphed, a subset of those typed in, and the end points for the graph.

A table of X and Y values corresponding to the letters on the grid appears below in the graph.

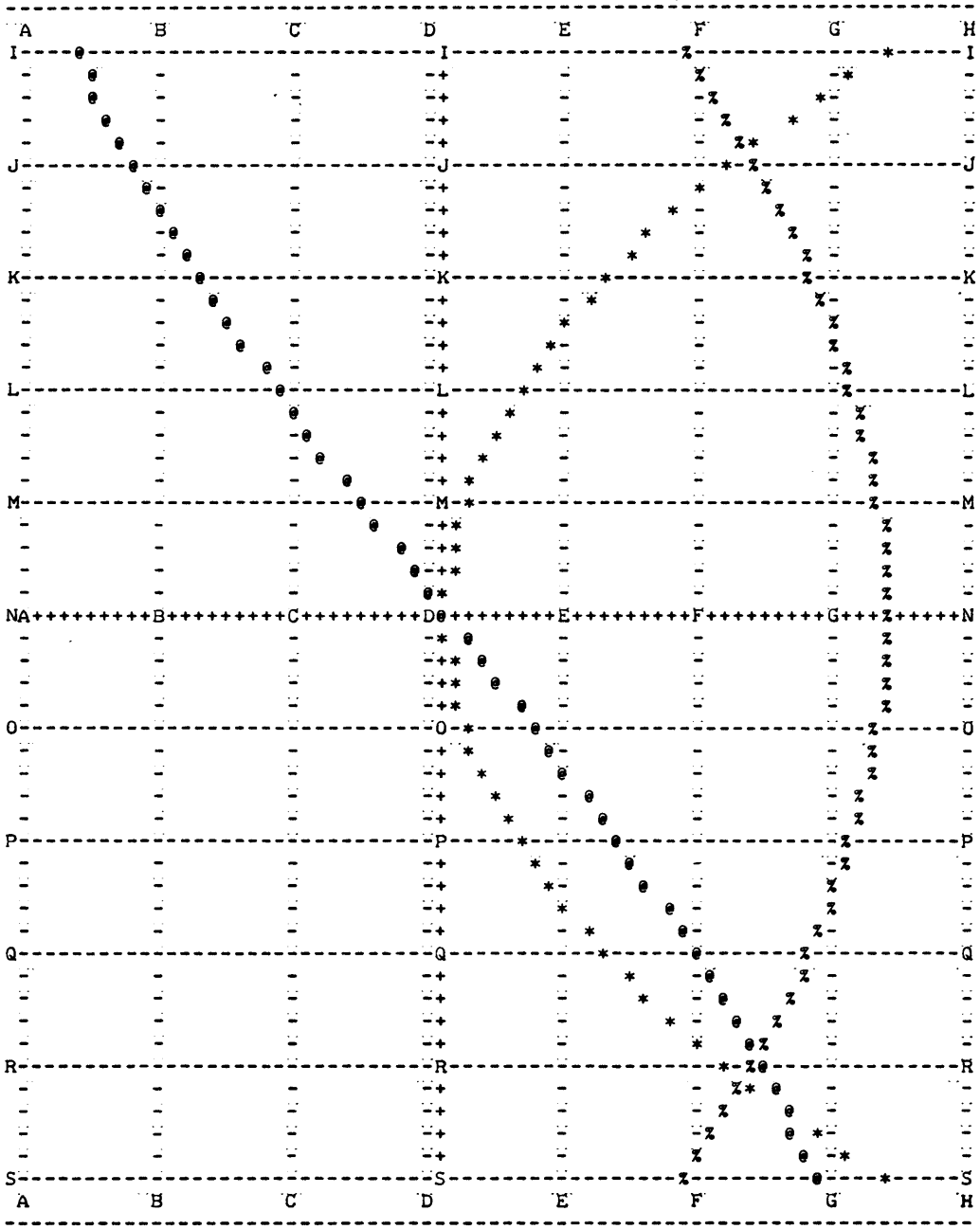
SPECIAL CONSIDERATIONS:

Any function which, at the beginning of the interval, evaluates to 5.1413 E-30 will be regarded as not typed in.

ACKNOWLEDGEMENTS: William J. Shipley
Blackburn College

RUN

```
10 DEF FNA(X)=X↑2
20 DEF FNB(X)=SIN(X)
30 DEF FNC(X)=COS(X)
RUN
FGRAPH

TYPE IN THE LAST LETTERS OF THE FUNCTIONS YOU WANT GRAPHED
SEPARATED BY COMMAS.  EXAMPLE: B,C,E WILL GRAPH FNB, FNC, AND FNE.
?A,B,C
INPUT THE BEGINNING AND END POINTS FOR THE GRAPH?-1,1
```



X VALUES	Y VALUES	FUNCTION KEY
I: -1	A: -.964236	*--> FNA
J: -.8	B: -.657324	@--> FNB
K: -.6	C: -.350412	%--> FNC
L: -.4	D: -4.35002E-02	
M: -.2	E: .263412	
N: -1.78814E-07	F: .570323	
O: .2	G: .877235	
P: .4	H: 1.18415	
Q: .6		
R: .8		
S: 1.		

```
DO YOU WANT TO TRY NEW END POINTS?NO
DO YOU WANT TO TRY A DIFFERENT COMBINATION OF THE FUNCTIONS THAT
YOU TYPED?NO

DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE:

PAPER TAPE FILE DUMP

FILDUM
36008

DESCRIPTION:

This program dumps the contents of files onto paper tape. It is intended for use with its companion program FILREA (HP 36011).

INSTRUCTIONS:

Before running the program declare the files to be dumped in line 8900

```
8900 FILES FIL1, FIL2 ....
```

in the order they are to be dumped. Also make sure there is sufficient leader on the paper tape. Do not have the punch on when beginning to run the program. When the program is run it will ask for the number of files declared. After inputting the number followed by a carriage return, the program will ask the user if the dump should stop at the first end-of-file. After the user responds with "Y" for yes, or "N" for no, he will be requested to turn on the tape punch. The program will then dump the files onto tape.

The characters "Control Q", "Control Shift N", and "Control Shift O" are special string control characters and should be avoided in the file data.

SPECIAL CONSIDERATIONS:

The program asks the user if he is on an HP 2000A, B, C, E, or F because the number of records per file varies accordingly.

Special numeric characters are dumped with the file data which indicates the data type of the next element; alpha, numeric, EOF, or EOR. These characters are ignored by FILREA on input.

ACKNOWLEDGEMENTS:

Joel F. Bartlett
Hewlett-Packard/Data Systems

```
RUN

GET-FILDUM
8900 FILES F1
RUN
FILDUM

IS T/S AN HP 2000 'A', 'B', 'C', 'E', OR 'F'?C
HOW MANY FILES ARE TO BE DUMPED?1
STOP DUMP OF FILES AT THE FIRST EOF (Y OR N)?Y


TURN ON THE TAPE PUNCH PLEASE (LEADER OK?).
 1
 2
 2
1  22  1  10  1  350100.  1  422505.  1  100
 2
THIS IS A SAMPLE FILE
 2
1  1  4  1  12  1  350300.  1  422503.  1  200
 2
USED TO DEMONSTRATE THE FILE UTILITY CONTRIBUTED PROGRAMS
 4
 3

DONE
```

TITLE: FILE MANIPULATION - CREATES, SORTS, UPDATES, COPIES, CHANGES FORMAT

FILES
36645

DESCRIPTION:

FILES is a complete file manipulation program offering the user 11 options including creating, sorting, updating, copying and changing formats of files.

INSTRUCTIONS:

GENERAL INFORMATION

A. The program requires a two record file named "SWAP".

B. The routines "CREATE" and "SORT" use files only in the random format, all other routines use either random or serial format.

C. All files to be used with this program must be "OPEN"ed before "RUN"ing.

ROUTINE SELECTION

A. Upon initially "RUN"ing the program, the following list will be output:

```
"1)...CREATE A FILE"
"2)...SORT A FILE"
"3)...LIST A FILE"
"4)...COPY A FILE"
"5)...X-PUNCH A FILE"
"6)...LIST THE FILE FORMAT"
"7)...UPDATE A RECORD IN A FILE"
"8)...ADD A FIELD TO A RECORD"
"9)...CHANGE A FIELD FROM A RECORD"
"10)..CHANGE FROM A RANDOM FILE TO A SERIAL FILE"
"11)..CHANGE FROM A SERIAL FILE TO A RANDOM FILE"
"12)..END THE PROGRAM"
```

Select the routine number of your choice (if the number entered is <1 or >12 the "YOUR CHOICE?" message will repeat.

continued on following page

SPECIAL CONSIDERATIONS:

'FILES' uses a two record work file named 'SWAP'. The program checks to see if this file exists or is in use at run time. If the user cannot gain exclusive access to 'SWAP' at run time the program will issue the message: "WORK FILE IN USE ENTER NAME OF 2 RECORD WORK FILE?"

Enter the name of any open, unused scratch file of at least 2 records in length. If this file does not meet the exclusive access criteria the "WORK FILE.... etc." message will be repeated.

ACKNOWLEDGEMENTS:

Terry von Gease
HP, Data Systems Division

INSTRUCTIONS continued

NOTE: This list is only output once at the start of the program; from then on only the lines:

"ENTER THE NUMBER OF THE ROUTINE TO BE RUN"
"YOUR CHOICE?"

will be output when any selected routine either ends or aborts.

"CREATE A FILE"

A. Prompting messages

1. "OUTPUT FILE NAME?" - enter the name of the file to be created.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "RECORD FORMAT = ?" - enter the record format according to the following conventions:

   a. "$" = Character string
   b. "#" = Numeric value
   c. "C$" = Character string constant (up to 5 string constants may be used)
   d. "C#" = Numeric constant (up to 5 numeric constant may be used)

   For example:

   By responding to "RECORD FORMAT = ?" with "$$C$#C$C$" a file will be created with records containing 6 fields in the following format:

   FIELD #1 - Character string #1
   FIELD #2 = Character string #2
   FIELD #3 = Character string constant #1
   FIELD #4 = Numeric value #1
   FIELD #5 = Character string constant #2
   FIELD #6 = Numeric constant #1

5. "$ CONSTANT n = ?" - if one or more character string constants were specified in the "RECORD FORMAT = ?", then enter the constant value (this message will be repeated for each string constant specified).

6. "# CONSTANT n = ?" - if one or more numeric constants were specified in the "RECORD FORMAT = ?", then enter the constant value (this message will be repeated for each numeric constant specified).

7. "** RECORD n ?" - n = what record number is about to be written. Enter a carriage return ("CR") to continue or a "\" (back slash) to end this routine (this message will be printed at the start of every record). Note that the "\" response will cause an "EOF" (end of file) to be printed at record n.

8. "# nn = ?" or "$ nn = ?" - if one or more string and/or numeric values were specified n "RECORD FORMAT = ?", then enter the appropriate values (these messages will print in the same sequence as they were entered in the "RECORD FORMAT = ?" and will be repeated for each record written.)

9. Example:

Message	Response
OUTPUT FILE NAME?	TEST
FILE USE A MASK?	NO
RECORD FORMAT = ?	$#C$C$C#
$ CONSTANT 1 = ?	STRING #1
$ CONSTANT 2 = ?	STRING #2
# CONSTANT 1 = $	3.14159
** RECORD 1 ?	(Carriage Return)
$ 1 = ?	TEST RECORD 1
# 1 = ?	11.11
** RECORD 2 ?	(Carriage Return)
$ 1 = ?	TEST RECORD #2
# 1 = ?	22.22
## RECORD 3 ?	\

INSTRUCTIONS continued

This will produce a two record file ("TEST") that will look like:

```
Record 1,    FIELD 1 = "TEST RECORD # 1"
             FIELD 2 = "11.11"
             FIELD 3 = "STRING #1"
             FIELD 4 = "STRING #2"
             FIELD 5 = "3.14159"
Record 2,    FIELD 1 = "TEST RECORD #2"
             FIELD 2 = "22.22"
             FIELD 3 = "STRING #1"
             FIELD 4 = "STRING #2"
             FIELD 5 = "3.14159"
Record 3,           = E O F
```

B. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - self explanatory, will cause the "OUTPUT FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters causes the request responsible to be repeated.

3. "c IS AN INVALID FORMAT CHARACTER" - if a character in the record format word is not "$", "#" or a "C" followed by a "$" or "#" will cause the "RECORD FORMAT = ?" request to be repeated.

4. "LAST CHARACTER IN FORMAT WORD MUST NOT BE C" - self explanatory - will cause the "RECORD FORMAT = ? request to be repeated.

5. "ONLY 5 (NUMBER)/(STRING) CONSTANTS ALLOWED" - self explanatory, will cause the "RECORD FORMAT = ?" request to be repeated.

6. "MORE THAN 256 WORDS OF STORAGE REQUIRED FOR RECORD n" - the physical record limits have been exceeded, and "EOF" (end of file) is printed in this record and the "CREATE" routine is ended.

7. "OUTPUT FILE IS FULL" - the physical limits of the file have been exceeded, the "CREATE" routine is ended.

"SORT A FILE"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to be sorted.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES" then enter the security mask (up to 6 characters).

4. "OUTPUT FILE NAME?" - enter the name of the file to be written with sorted data.

5. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

6. "WHAT IS THE MASK?" - if the response to #5 was "YES" then enter the security mask (up to 6 characters).

7. "INPUT FILE STARTING AND ENDING RECORD # ?" - enter the starting and ending record numbers to be sorted. (To sort an entire file enter 1, n. n = a number equal to or larger than the logical length of the input file + the starting record #). Note that the program will process from the starting record # thru either the ending record # or the first end of file (EOF) mark encountered.

8. "SORT ON FIELD # ?" - enter the field number (1 thru nn) to be sorted on.

9. "IS FIELD nn A CHARACTER STRING?" - enter "YES" or "NO" as the case may be ("Y" or "N").

10. "ENTER THE CHARACTER STRING STARTING AND ENDING CHARACTER POSITIONS OR ENTER 0, 0 TO USE THE ENTIRE STRING?" - if the response to #9 was "YES" then by entering the proper starting and ending characte positions a sort may be done on partial portion of a character string. Note that only the starting position must be within the physical limits of the string. Also note that if the response to #9 was "NO" and the field in question is a character string, the program will default and sort on the entire string.

INSTRUCTIONS continued

11. "ASCENDING OR DESCENDING SEQUENCE (0 OR 1)?" - enter your choice.

B. Information Messages

1. "nnn RECORDS SORTED" - signals that sort routine is ended.

C. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input or output is either in use, protected, or non-existent. Causes the "(OUTPUT) (INPUT)" FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "INPUT AND OUTPUT CANNOT BE THE SAME FILE" - causes the "OUTPUT FILE NAME?" request to be repeated.

4. "UPPER LIMIT LESS THAN LOWER LIMIT" - causes the "INPUT FILE STARTING AND ENDING RECORD #?" request to be repeated.

5. "STARTING RECORD BEYOND FILE LIMITS" - the input file starting record number is beyond the physical length of the file. Causes the "INPUT FILE STARTING AND ENDING RECORD #?" request to be repeated.

6. "STARTING CHARACTER POSITION IS LARGER THAN ENDING OR 0" - issued when an improper response is given to the string starting and ending character position request. Causes the "ENTER THE CHARACTER STRING STARTING AND ENDING CHARACTER POSITIONS OR ENTER 0, 0 TO USE THE ENTIRE STRING?" request to be repeated.

7. "STRING BEING SORTED ON IS ONLY nn CHARACTERS LONG, THE SPECIFIED STARTING CHARACTER POSITION IS BEYOND THE LIMITS OF THE STRING" - self explanatory, causes the sort routine to end.

8. "SORT INPUT FILE RECORD nn IS NOT A NUMBER OR STRING AS SPECIFIED" - the field specified for sort control is not always a character string or always a number value. Causes the sort routine to end.

9. "OUTPUT FILE IS FULL" - this message will be issued if the file specified for sort output has a physical length equal to or smaller than the logical length (starting record thru ending record) of the input file. Causes the sort routine to end. Note that the output file is sorted and complete with all the records input up to this point.

10. "NOT ENOUGH FIELDS IN RECORD nnn" - the value specified in the "SORT ON FIELD #?" request is too large for this record. Causes the sort routine to abort.

"LIST A FILE"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to be listed.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES" then enter the security mask (up to 6 characters).

4. "INPUT FILE STARTING AND ENDING RECORD #?" - enter the starting and ending record numbers to be listed. (To list an entire file enter 1,n. n = a number equal to or larger than the logical length of the input file + the starting record #). Note that the input file is listed from the starting record number thru either the ending record number or first End of File (EOF) marks encountered.

5. "*** LINE PRINTER?" - if the line printer is required for output, enter a control W (W^C) and a carriage return (CR). If the line printer is not desired, merely enter enter a carriage return (CR) to continue.

INSTRUCTIONS continued

B. Sample Printout

1. If file created as an example in "CREATE A FILE" were to be listed using this routine, the results would be thus.

```
*** RECORD 1
TEST RECORD #1
#11.11
STRING #1
STRING #2
#3.14150
   EOR
*** RECORD 2
TEST RECORD #2
#22.22
STRING #1
STRING #2
#3.14159
   EOR
*** RECORD 3
   EOF
```

C. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input is either protected or non-existent. Causes the "INPUT FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "UPPER LIMIT LESS THAN LOWER LIMIT" - causes the "INPUT FILE STARTING AND ENDING RECORD #?" request to be repeated.

4. "STARTING RECORD BEYOND FILE LIMITS" - the input file starting record number is beyond the physical length of the file. Causes the "INPUT FILE STARTING AND ENDING RECORD #?" request to be repeated.

"COPY A FILE"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to be copied.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "OUTPUT FILE NAME?" - enter the name of the file to be written.

5. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

6. "WHAT IS THE MASK?" - if the response to #5 was "YES", then enter the security mask (up to 6 characters.

7. "INPUT FILE STARTING AND ENDING RECORD #?" - enter the starting and ending record numbers to be copied. (To copy an entire file, enter 1,n. n = a number equal to or larger than the logical length of the input file + the starting record #). Note that the program will process from the starting record number thru either the ending record number or the first End of File (EOF) mark encountered.

8. "OUTPUT FILE STARTING RECORD #?" - self explanatory.

B. Information Messages

1. "nnn RECORDS TRANSFERRED" - signals that the file copy has been successfully executed and the copy routine is ended.

C. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input or output is either in use, protected, or non-existent. Causes the "($\frac{\text{OUTPUT}}{\text{INPUT}}$) FILE NAME?" request to be repeated.

INSTRUCTIONS continued

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "INPUT AND OUTPUT CANNOT BE THE SAME FILE" - causes the "OUTPUT FILE NAME?" request to be repeated.

4. "UPPER LIMIT LESS THAN LOWER LIMIT" - causes the "INPUT FILE STARTING AND ENDING RECORD #?" request to be repeated.

5. "STARTING RECORD BEYOND FILE LIMITS" - the input file starting record number is beyond the physical length of the file. Causes the "INPUT FILE STARTING AND ENDING RECORD #?" or "OUTPUT FILE STARTING RECORD #?" request to be repeated.

6. "FILE BEING WRITTEN IS TOO SMALL" - causes the copy routine to end.

"X-PUNCH A FILE"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to be punched.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "INPUT FILE STARTING AND ENDING RECORD #?" - enter the starting and ending record numbers to be punched. (To sort an entire file enter 1,n. n = a number equal to or larger than the logical length of the input file + the starting record #). Note that the program processes from the starting record number thru either the ending record number or the first End of File (EOF) encountered.

5. "YOU HAVE 10 SECONDS TO MAKE THE TAPE PUNCH READY" - exactly that.

B. Output format - each field of each record is output with an "X-OFF" and carriage return (CR) immediately following it.

C. Information Messages

1. "nnn RECORDS PUNCHED" - signals that the punch routine is ended.

D. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input is either protected or non-existent. Causes the "INPUT FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

"DETERMINE THE FORMAT OF A FILE"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to be listed.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

B. Sample output using the file built in "CREATE A FILE" assuming the physical length of the file "TEST" to be 10 records.

```
1 to 2 DATA
3 to 10  END OF FILE
    10  PHYSICAL END OF FILE
```

C. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input is either protected or non-existent. Causes the "INPUT FILE NAME?" request to be repeated.

INSTRUCTIONS continued

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

"UPDATE A RECORD IN A FILE"

A. Prompting Messages

1. "OUTPUT FILE NAME?" - enter the name of the file to be updated.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "ENTER RECORD #, FIELD #, NUMBER (0) OR STRING (1)"
("ENTER 0, 0, 0 TO END THE UPDATE ROUTINE")? - the record # and field # requests are self-explanatory, the number or string request refers to the data to be entered as update information not the original configuration of the field number specified. Note that the second line of this message is printed only once, but the response of 0, 0, 0, is always a valid response to the request. (0, 0, 0, is the only way to end the update routine.)

5. "R # nnn, F # nn, NEW ($\begin{matrix}\text{NUMBER}\\\text{STRING}\end{matrix}$) VALUE = ?" - enter the appropriate update information (R = record, F = field). After entering the update data, the first line of message #4 will be issued.

B. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input is either protected or non-existent. Causes the ""INPUT FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "RECORD # nnn IS BEYOND THE FILE LIMITS" - causes the "ENTER RECORD #", FIELD #, NUMBER (0) or STRING (1)?" request to be repeated.

4. "FIELD # MUST BE GREATER THAN 0, RE-ENTER?" - enter only the field #.

5. "ENTER 0 FOR A NUMBER OR 1 FOR A STRING?" - issued if the number or string specification is not 0 or 1 - re-enter the number or string specification at this time.

6. "NOTE ENOUGH FIELDS FOR THE REQUESTED UPDATE" - causes the "ENTER RECORD #, FIELD #, NUMBER (0) or STRING (1)?" request to be repeated.

7. "MORE THAN 256 WORDS OF STORAGE REQUIRED FOR THIS RECORD" - causes the "ENTER RECORD #, FIELD #, NUMBER (0) or STRING (1)?" request to be repeated.

"ADD A FIELD TO A RECORD"

A. Prompting Messages

1. "OUTPUT FILE NAME?" - enter the name of the file to be updated.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "STARTING AND ENDING RECORD #?" - enter the starting record, ending record # - note that the program will process from the starting record # thru the ending record # or to the first End of File (EOF) mark encountered.

5. "ADD FIELD #?" - enter any value from 1 thru the number of fields in a record +1 - note that if a field is added between two existing fields (for example, the response to the prompt was 3 and the record has 4 fields) the new field will be inserted immediately behind the existing field #2 and the old field #3 will become field #4 and so forth thru the last field in the record.

INSTRUCTIONS continued

6. "NUMBER OR STRING (0 or 1)?" - enter the appropriate value.

7. "DO YOU WANT TO USE A CONSTANT VALUE?" - enter "YES" or "NO" as the case may be ("Y" or "N").

8. "ENTER THE (NUMERIC / STRING) CONSTANT?" - if the response to #7 was "YES", then enter the constant value to be added.

9. "R # nnn, F # nn, ADD (NUMBER / STRING) VALUE?" - if the response to #7 was "NO", then this message will be issued for each record specified in the starting and ending record statement.

B. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input is either protected or non-existent. Causes the "INPUT FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "FIELD # MUST BE GREATER THAN 0, RE-ENTER?" - self explanatory.

4. "ENTER 0 FOR A NUMBER OR 1 FOR A STRING?" - the response to "NUMBER OR STRING (0 OR 1)?" request was not 0 or 1. Re-enter the appropriate value.

5. "STARTING RECORD BEYOND THE FILE LIMITS" - causes the "STARTING AND ENDING RECORD #?" request to be repeated.

6. "STARTING RECORD GREATER THAN ENDING RECORD" - causes the "STARTING AND ENDING RECORD #?" request to be repeated.

7. "MORE THAN 256 WORDS OF STORAGE REQUIRED FOR RECORD" - causes the update routine to abort at this point. Note that all records updated to this point are still valid.

8. "NOT ENOUGH FIELDS FOR THE REQUESTED ADD" - causes the add routine to abort. Note that all records updated to this point are still valid.

9. "LAST RECORD IN FILE UPDATED" - information only, causes the add routine to end at this point.

"DELETE A FIELD FROM A RECORD"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to be updated.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "STARTING AND ENDING RECORD #?" - enter the starting record, ending record # - note that the program will process from the starting record # thru the ending record # or to the first End of File (EOF) mark encountered.

5. "DELETE FIELD #?" - enter any value from 1 thru the number of fields in a record.

B. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input is either protected or non-existent. Causes the "INPUT FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "FIELD # MUST BE GREATER THAN 0, RE-ENTER?" - self explanatory.

4. "NOT ENOUGH FIELDS FOR THE REQUESTED DELETE" - causes the delete routine to abort. Note that all records updated to this point are still valid.

5. "LAST RECORD IN FILE UPDATED" - information only, causes the add routine to end at this point.

INSTRUCTIONS continued

"CHANGE FROM A RANDOM FILE TO A SERIAL FILE"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to be restructured.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "OUTPUT FILE NAME?" - enter the name of the file to be written.

5. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

6. "WHAT IS THE MASK?" - if the response to #5 was "YES", then enter the security mask (up to 6 characters).

7. "INPUT FILE STARTING AND ENDING RECORD #?" - enter the starting and ending record numbers to be converted. (To convert an entire file, enter 1,n. n = a number equal to or larger than the logical length of the input file + the starting record #). Note that the program will process from the starting record number thru either the ending record number or the first End of File (EOF) mark encountered.

8. "OUTPUT FILE STARTING RECORD #?" - self explanatory.

B. Information Messages

1. "nnn RECORDS CONVERTED FROM RANDOM TO SERIAL FORMAT" - signals the end of the routine.

C. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input or output is either in use, protected, or non-existent. Causes the (OUTPUT / INPUT) FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED" - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "INPUT AND OUTPUT CANNOT BE THE SAME FILE" - causes the "OUTPUT FILE NAME?" request to be repeated.

4. "UPPER LIMIT LESS THAN LOWER LIMIT" - causes the "INPUT FILE STARTING AND ENDING RECORD #?" request to be repeated.

5. "STARTING RECORD BEYOND FILE LIMITS" - the input file starting record number is beyond the physical length of the file. Causes the "INPUT FILE STARTING AND ENDING RECORD #?" or "OUTPUT FILE STARTING RECORD #?" request to be repeated.

6. "OUTPUT FILE TO SMALL, ONLY nnn RECORDS CONVERTED" - causes the routine to end.

"CHANGE FROM A SERIAL FILE TO A RANDOM FILE"

A. Prompting Messages

1. "INPUT FILE NAME?" - enter the name of the file to restructured.

2. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

3. "WHAT IS THE MASK?" - if the response to #2 was "YES", then enter the security mask (up to 6 characters).

4. "OUTPUT FILE NAME?" - enter the name of the file to be written.

5. "FILE USE A MASK?" - enter "YES" or "NO" as the case may be ("Y" or "N").

6. "WHAT IS THE MASK?" - if the response to #5 was "YES", then enter the security mask (up to 6 characters).

7. "INPUT FILE STARTING RECORD #?" - enter the starting record # of the file to be restructured.

INSTRUCTIONS continued

8. "HOW MANY FIELDS IN ONE LOGICAL RECORD?" - enter the number of fields desired in one random format record. (The total number of fields in the serial input file does not have to be an even multiple of the number of fields in one random record. The residual fields, if any, will be in the last random record.)

B. Information Messages

1. "nnn RECORDS CREATED" - signals the end of this routine.

C. Error Messages

1. "REQUESTED FILE UNAVAILABLE" - the file specified for input or output is either in use, protected, or non-existent. Causes the "(OUTPUT / INPUT) FILE NAME?" request to be repeated.

2. "ONLY 6 CHARACTERS ALLOWED: - issued if a file name or file security mask was entered longer than 6 characters. Causes the request responsible to be repeated.

3. "INPUT AND OUTPUT CANNOT BE THE SAME FILE" - causes the "OUTPUT FILE NAME?" request to be repeated.

4. "STARTING RECORD BEYOND FILE LIMITS" - causes the "INPUT FILE STARTING RECORD #?" request to repeat.

5. "OUTPUT FILE TOO SMALL, ONLY nnn RECORDS CREATED" - causes the routine to end.

6. "TOO MANY CHARACTERS FOR ONE RECORD" - more than 256 words of storage are required for a record. Causes the routine to end.

```
RUN

RUN
FILES


INSTRUCTIONS ?Y
1)...CREATE A FILE
2)...SORT A FILE
3)...LIST A FILE
4)...COPY A FILE
5)...X-PUNCH A FILE
6)...LIST THE FILE FORMAT
7)...UPDATE A RECORD IN A FILE
8)...ADD A FIELD TO A RECORD
9)...DELETE A FIELD FROM A RECORD
10)..CHANGE FROM A RANDOM FILE TO A SERIAL FILE
11)..CHANGE FROM A SERIAL FILE TO A RANDOM FILE
12)..END THE PROGRAM

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?1

OUTPUT FILE NAME ?TEST

FILE USE A MASK?NO

RECORD FORMAT = ?$"C$C$C"\
$#C$C$C#

$ CONSTANT 1 = ?STRING #1

$ CONSTANT 2 = ?STRING #2

# CONSTANT 1 = ?3.14159

** RECORD   1 ?
$  1 = ?TEST RECORD 1
#  1 = ?11.11

** RECORD   2 ?
$  1 = ?TEST RECORD #2
#  1 = ?22.22

OUTPUT FILE IS FULL

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?3
```

```
INPUT FILE NAME ?TEST

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

*** LINE PRINTER?NO

*** RECORD   1
TEST RECORD 1
# 11.11
STRING #1
STRING #2
# 3.14159
EOR

*** RECORD   2
TEST RECORD #2
# 22.22
STRING #1
STRING #2
# 3.14159
EOF


ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?2

INPUT FILE NAME ?TEST

FILE USE A MASK?NO

OUTPUT FILE NAME ?TER1

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

SORT ON FIELD #?2

IS FIELD # 2 A CHARACTER STRING?Y

ENTER THE CHARACTER STRING STARTING AND ENDING CHARACTER POSITIONS
OR ENTER 0,0 TO USE THE ENTIRE STRING?0,0

ASCENDING OR DESCENDING SEQUENCE (0 OR 1)?1

  1 RECORDS SORTED

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?3

INPUT FILE NAME ?TER1

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

*** LINE PRINTER?NO

*** RECORD   1
TEST RECORD #2
# 22.22
STRING #1
STRING #2
# 3.14159
EOR

*** RECORD   2
TEST RECORD 1
# 11.11
STRING #1
STRING #2
# 3.14159
EOR
```

```
ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?4

INPUT FILE NAME ?TER1

FILE USE A MASK?NO

OUTPUT FILE NAME ?TER2

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

OUTPUT FILE STARTING RECORD #?3
  2 RECORDS TRANSFERED


ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?5

INPUT FILE NAME ?TER2

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

LOWER LIMIT IS LARGER THAN THE FILE

INPUT FILE STARTING AND ENDING RECORD #?2,1

UPPER LIMIT LESS THAN LOWER LIMIT

INPUT FILE STARTING AND ENDING RECORD #?3,4

YOU HAVE 10 SECONDS TO MAKE THE TAPE PUNCH READY
TEST RECORD #2
 22.22
STRING #1
STRING #2
 3.14159
TEST RECORD 1
 11.11
STRING #1
STRING #2
 3.14159

  2 RECORDS PUNCHED

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?6

INPUT FILE NAME ?TEST

FILE USE A MASK?NO
  1 TO   2 DATA
         2 PHYSICAL END OF FILE

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?7

OUTPUT FILE NAME ?TEST

FILE USE A MASK?NO

ENTER RECORD #, FIELD #, NUMBER (0) OR STRING (1)
(ENTER 0,0,0 TO END THE UPDATE ROUTINE)?1,5,0

R #  1, F # 5, NEW NUMBER VALUE =?3.1415927

ENTER RECORD #, FIELD #, NUMBER (0) OR STRING (1)?0,0
??0

ENTER THE NUMBER OF THE ROUTINE TO BE RUN
```

```
YOUR CHOICE?8

OUTPUT FILE NAME ?TEST

FILE USE A MASK?NO

STARTING AND ENDING RECORD #?1,2

ADD FIELD #?2

NUMBER OR STRING (0 OR 1)?0

DO YOU WANT TO USE A CONSTANT VALUE?Y

ENTER THE NUMERIC CONSTANT?12358

LAST RECORD IN FILE UPDATED

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?3

INPUT FILE NAME ?TEST

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

*** LINE PRINTER?NO

*** RECORD   1
TEST RECORD 1
# 12358
# 11.11
STRING #1
STRING #2
# 3.14159
EOR

*** RECORD   2
TEST RECORD #2
# 12358
# 22.22
STRING #1
STRING #2
# 3.14159
EOF


ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?9

OUTPUT FILE NAME ?TEST

FILE USE A MASK?NO

STARTING AND ENDING RECORD #?1,2

DELETE FIELD #?4

LAST RECORD IN FILE UPDATED

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?3

INPUT FILE NAME ?TEST

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

*** LINE PRINTER?NO

*** RECORD   1
TEST RECORD 1
# 12358
```

```
# 11.11
STRING #2
# 3.14159
EOR

*** RECORD   2
TEST RECORD #2
# 12358
# 22.22
STRING #2
# 3.14159
EOF


ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?10

INPUT FILE NAME ?TEST

FILE USE A MASK?NO

OUTPUT FILE NAME ?TER3

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?1,2

OUTPUT FILE STARTING RECORD #?5

  2 RECORDS CONVERTED FROM RANDOM TO SERIAL FORMAT

ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?3

INPUT FILE NAME ?TER3

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?5,10

*** LINE PRINTER?NO

*** RECORD   5
TEST RECORD 1
# 12358
# 11.11
STRING #2
# 3.14159
TEST RECORD #2
# 12358
# 22.22
STRING #2
# 3.14159
EOF


ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?11

INPUT FILE NAME ?TER3

FILE USE A MASK?NO

OUTPUT FILE NAME ?TER2

FILE USE A MASK?NO

INPUT FILE STARTING RECORD #?5

OUTPUT FILE STARTING RECORD #?6

HOW MANY FIELDS IN ONE LOGICAL RECORD ?6

  2 RECORDS CREATED
```

```
ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?3

INPUT FILE NAME ?TER2

FILE USE A MASK?NO

INPUT FILE STARTING AND ENDING RECORD #?6,10

*** LINE PRINTER?NO

*** RECORD    6
TEST RECORD 1
# 12358
# 11.11
STRING #2
# 3.14159
TEST RECORD #2
EOR

*** RECORD    7
# 12358
# 22.22
STRING #2
# 3.14159
EOF


ENTER THE NUMBER OF THE ROUTINE TO BE RUN

YOUR CHOICE?12

DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE: KEYBOARD FILE LOADING PROGRAM

FILIN
36007A

DESCRIPTION:

This program is used to load a file from the keyboard. The input consists of numbers or characters separated by commas up to 72 characters per line. Any input which conforms to one of the numeric formats is considered to be numeric.

INSTRUCTIONS:

A files statement must be included in the program which names the file to be loaded. This file must have been previously opened.

When RUN the program asks if you want instructions initially.

When inputting to the file, input up to 72 characters per line.

To terminate the program type "control C".

End follows ? prompt

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

Warren Nelson
Hewlett-Packard/Canada

```
RUN
OPEN-FILEIN,12
RUN
DONE
GET-FILIN
RUN
FILIN

DO YOU WANT INSTRUCTIONS?YES
FILIN IS USED TO LOAD A FILE FROM A KEYBOARD. BEFORE THE PROGRAM
IS RUN A FILES STATEMENT MUST BE INCLUDED WHICH NAMES THE FILE TO BE
LOADED. THIS FILE MUST HAVE BEEN PREVIOUSLY OPENED.
WHEN THE PROGRAM ASKS FOR INPUT, YOU MAY INPUT UP TO 72 CHARACTERS
PER LINE. THE INPUT SHOULD CONSIST OF NUMBERS OR CHARACTER STRINGS
SEPARATED BY COMMAS. ANY INPUT WHICH CONFORMS TO ONE OF THE STANDARD
NUMERIC FORMATS IS CONSIDERED TO BE NUMERIC.
TO TERMINATE THE PROGRAM, TYPE A 'CONTROL C' IN THE INPUT.
?
```

CONTRIBUTED PROGRAM **BASIC**

TITLE:

FILIS
FILE LISTING PROGRAM/INSTRUCTIONS 36272

DESCRIPTION:

This program is a file lister. The program "FILIS" contains the instructions for the program "FILISI" (which can be run directly).

INSTRUCTIONS:

Most of the needed instructions are contained in the program "FILIS". The only other information needed is a description of Aardvark and Company Writing Team format for printing strings:

Control letters are printed with an apostrophe (') over them:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Lower case letters are printed with a period (.) over them:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Characters with parity are darkened:
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Except space which is printed, when it has parity, as:
▯

When you request the format option:
At the beginning of each record, the word "FORMAT:" is printed followed by the format of that record in the form:

$	For a string
#	For a number
EOR	For end-of-record
EOF	For end-of-file

If a string is printed which has spaces at the end, the length of that string will be printed if the format option is requested.
The length appears in this form:
▪ Length (Length of string in characters)

A null string (a string containing 0 characters) will be printed as:
▪ Null String

SPECIAL CONSIDERATIONS:

The file "ASCII" must be open in the library and prepared by the program "ASCII%".

The Aardvark and Company Writing Team has designed programs to take up an absolute minimum of computer storage and perform a maximum purpose. The team encourages people to send good programs to Aardvark. As a slight encouragement, the team will give anyone who sends a program which is accepted a free "subscription" to the program handbook, and include the contributor as a member of the writing team.

ACKNOWLEDGEMENTS:

Philip J. Tubb
Aardvark and Company
2130 Bell Court
Lakewood, Colorado 80215

RUN

```
RUN
FILIS

DO YOU WANT INSTRUCTIONS?YES

EACH TIME I PRINT AN * YOU MAY INPUT AN OPTION LIST AND A FILE NAME.

AN OPTION LIST IS ANY COMBINATION OF THE LETTERS L,S,E,R,F, AND I
FOLLOWED BY A COMMA. HERE IS WHAT THOSE LETTERS MEAN:

L     INDICATES THAT THIS IS THE LAST FILE TO BE LISTED.
S     CAUSES THE LISTING TO CONTINUE TO THE NEXT FILE AFTER THE
      FIRST EOF IS ENCOUNTERED IN THIS FILE.
E     CAUSES THE PROGRAM TO SKIP RECORDS WHICH CONTAIN ONLY AN EOF MARK.
R     CAUSES THE PROGRAM TO SKIP RECORDS WHICH CONTAIN ONLY AN EOR MARK.
F     CAUSES THE FORMAT OF THE FILE TO BE PRINTED.
I     ALLOWS YOU TO INPUT WHICH RECORDS ARE TO BE LISTED. THE PROGRAM
      WILL PRINT  *** RECORD    AND ALLOW YOU TO INPUT A NUMBER.
      IT WILL LIST THAT RECORD, IF PRESENT, AND ASK FOR A RECORD
      NUMBER AGAIN.  IF YOU ARE FINISHED LISTING RECORDS IN THAT FILE,
      INPUT THE RECORD NUMBER AS 0.


A FILE NAME CONSISTS OF THE NAME OF THE FILE, THEN (OPTIONALLY) A
COMMA AND THE MASK TO BE USED.



HERE IS AN EXAMPLE OF THE INPUT WHICH WOULD BE USED TO LIST THE FILE
'MYFILE' USING THE MASK 'MYMASK' AND HAVING THE FORMAT PRINTED:

*LF,MYFILE,MYMASK

HERE IS THE SAME EXAMPLE, EXCEPT WITH NO MASK:

*LF,MYFILE



IF IT PRINTS AN * AND YOU HAVE NO MORE FILES TO BE LISTED, PUSH RETURN.


*F,ASCII
*FIL,NAME




*** FILE 1


*** RECORD 1

FORMAT:
$$ EOR

@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑← !"#$%_'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
EOR

*** RECORD 2

FORMAT:
$$ EOF

@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←!"#$%_'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑←
EOF


*** FILE 2
```

```
*** RECORD 2

FORMAT:
$$$$ EOR

BARTLETT,MARY
SE
ESR
NOV
EOR

*** RECORD 3

FORMAT:
$$$$ EOR

BILLINGS,IRV
SA
ESR
NOV
EOR

*** RECORD 0




DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE: LISTS FILE CONTENTS BY RECORD NUMBER

FILIST
36009

DESCRIPTION:

This program will list the contents of any file or files on the teleprinter.

INSTRUCTIONS:

Before running this program declare your files in line 8900. For example:

```
8900 FILES BFILE, BUSØØ
```

Before each new listing a user has the option to stop listing at the first EOF (YES, NO or QUIT). Multiple files may be listed if they have been declared in line 8900. Running the program will then tell you the file number (1,2,3,etc.), record number, and contents for all the declared files.

The program issues a request for a file name - end terminates the program.

The program requests a random y/n input. If Y, the program requests a record number. If N the program requests if first EOF should stop listing (Y, N, or Q)

If Y, the file is listed to the first E.O.F. If N, the complete file is listed. If Q the program restarts at file name request.

ACKNOWLEDGEMENTS:

Joel F. Bartlett
Hewlett-Packard/Data Systems

```
RUN


GET-FILIST
8900 FILES F1
RUN
FILIST


STOP LISTING FILE 1     AT THE FIRST EOF (Y OR N OR Q)?Y


FILE 1     RECORD 1

1  22  1  10  1  350100.  1  422505.  1  100 THIS IS A SAMPLE FILE
1  1  4  1  12  1  350300.  1  422503.  1  200
USED TO DEMONSTRATE THE FILE UTILITY CONTRIBUTED PROGRAMS

FILE 1     RECORD 2

END OF FILE 1

STOP LISTING FILE 2     AT THE FIRST EOF (Y OR N OR Q)?Q

DONE
```

CONTRIBUTED PROGRAM **BASIC**

FILMAN
36006

TITLE:

FILE MANAGER

DESCRIPTION:

This program aids in maintaining data files for time-sharing. Files must be structured such that all logical records have identical data types.

OPERATIONS: HELP
LIST
MODIFY
CREATE
DESTROY
TAPE LOAD
TAPE DUMP

INSTRUCTIONS:

After loading or getting FILMAN, type 1 FILES filename. Then type RUN.

OPERATIONS: HELP
LIST
MODIFY
CREATE
DESTROY
TAPE
STOP

WHICH OPERATION? HELP
HELP

'LIST' READS THE SPECIFIED ELEMENT(S) FROM THE SPECIFIED RECORD(S)
'MODIFY' CHANGES THE SPECIFIED ELEMENT(S) OF THE SPECIFIED RECORD(S)
'CREATE' WRITES ALL ELEMENTS INTO EACH RECORD
'DESTROY' ERASES ALL ELEMENTS OF THE SPECIFIED RECORD
'TAPE' LOADS FROM OR DUMPS ONTO PAPER TAPE
'STOP' HALTS EXECUTION

'LIST' AND 'MODIFY' REQUEST A RECORD NUMBER AND AN ELEMENT NUMBER. 'Ø' RESPONSE SPECIFIES ALL RECORDS OR ELEMENTS.

'9999' RESPONSE TO ANY OPERATION REQUEST ABORTS THE OPERATION

WHICH OPERATION?

SPECIAL CONSIDERATIONS:

Logical records must be stored one per physical record (64 words). Maximum number of strings is 10. Maximum number of numeric elements is 32. The paper tape load routine will only load tapes previously created under the paper tape dump routine of FILMAN.

ACKNOWLEDGEMENTS:

George Schapiro
Hewlett-Packard/Cupertino

RUN

```
GET-$FILMAN
1 FILES BUSØØ
RUN
FILMAN

** FILE MAINTENANCE **

FILE HAS 11   RECORDS AVAILABLE
 11   ARE USED
NEW OR OLD FILE?NEW

DESCRIBE RECORD STRUCTURE
TO CORRECT PREVIOUS MISTAKES, INPUT 9999

HOW MANY ELEMENTS IN EACH RECORD?6
FOR EACH ELEMENT, INPUT ITS TYPE:
          A - ALPHANUMERIC STRING
          N - NUMBER
# 1    ?N
# 2    ?N
# 3    ?N
# 4    ?N
# 5    ?N
# 6    ?N

OPERATIONS:     HELP
                LIST
                MODIFY
                CREATE
                DESTROY
                TAPE
                STOP

WHICH OPERATION?LIST
LIST
RECORD #, ELEMENT #?1,4
RECORD # 1
# 4     3
END OF LIST

WHICH OPERATION?LIST
LIST
RECORD #, ELEMENT #?3,1
RECORD # 3
# 1     5296.5
END OF LIST

WHICH OPERATION?MODIFY
MODIFY
RECORD #, ELEMENT #?3,1
RECORD # 3
# 1    ?5ØØØ.5
END OF MODIFY

WHICH OPERATION?LIST
LIST
RECORD #, ELEMENT #?3,1
RECORD # 3
# 1     5ØØØ.5
END OF LIST

WHICH OPERATION?CREATE
CREATE
# OF FIRST RECORD TO BE WRITTEN?3
RECORD # 3
# 1    ?1
# 2    ?2
# 3    ?3
# 4    ?4
# 5    ?5
# 6    ?6
RECORD # 4
# 1    ?
```

CONTRIBUTED PROGRAM **BASIC**

TITLE: LOADS A FILE FROM THE TELETYPE

FILOAD
36010

DESCRIPTION:

This program allows keyboard entry of data into a file. Either numeric or string data can be entered without specifying type.

INSTRUCTIONS:

Before running the program declare the file in line 8900 by:

8900 FILES ...

Each time the program types a question mark (?) enter the next data item, either string or numeric, followed by a carriage return.

The next record will be started either when the preceding one cannot accept the next data item or the data input is a single "Up Arrow" (↑). In the latter case the arrow is not stored, but the program advances to the next record.

To stop the program before end-of-file just ~~input a "Control C"~~. Press Break

SPECIAL CONSIDERATIONS:

The case where a number is to be input as a string requires that some non-numeric character be included for the identification by the program as a string. The backspace arrow (←) should be avoided as it causes unpredictable results. If an error is made type alt mode/esc. and re-enter the data item.

The contents of an old file may be updated by record by bypassing earlier records with an "↑". The contents of earlier records will not be disturbed in the process.

The program asks if the user is on an HP 2000A, B, C, E, or F Time-Share System because the size of a record will vary from 64 words on the 2000A, 64 words on the 2000B, 256 words on the 2000C, 129 words on the 2000E, and 256 words on the 2000F.

ACKNOWLEDGEMENTS:

```
RUN

GET-FILOAD
OPEN-F1,2
8900 FILES F1
RUN
FILOAD

IS T/S AN HP 2000 'A','B', 'C', 'E', OR 'F'?C

STARTED RECORD # 1
?1  22  1  10  1  350100.  1  422505.  1  100
?THIS IS A SAMPLE FILE
?1  1  4  1  12  1  350300.  1  422503.  1  200
?USED TO DEMONSTRATE THE FILE UTILITY CONTRIBUTED PROGRAMS
?
DONE



GET-FILIST
8900 FILES F1
RUN
FILIST

IS T/S AN HP 2000'A', 'B', 'C', 'E', OR 'F'?C

STOP LISTING FILE 1    AT THE FIRST EOF (Y OR N OR Q)?Y


FILE 1    RECORD 1

1  22  1  10  1  350100.  1  422505.  1  100 THIS IS A SAMPLE FILE
1  1  4  1  12  1  350300.  1  422503.  1  200
USED TO DEMONSTRATE THE FILE UTILITY CONTRIBUTED PROGRAMS

FILE 1    RECORD 2

END OF FILE 1

STOP LISTING FILE 2    AT THE FIRST EOF (Y OR N OR Q)?Q

DONE
```

FILREA
36011

TITLE: REENTERS THE DATA TAPE DUMPED BY FILDUM

DESCRIPTION:

This program will load files from paper tape prepared by the program FILDUM. Using these two programs together preserves file structure.

INSTRUCTIONS:

Before running the program declare the files in line 8900 by:

8900 FILES FILE 1, FILE 2,...

in the sequence in which they are to receive data.

When the program is run it will request the number of files declared and then ask for the paper tape reader to be turned on. The files will then be loaded from the paper tape.

SPECIAL CONSIDERATIONS:

This program requires automatic reader control.

"NEED # FILES, YOU ARE # SHORT.", implies the paper tape requires more files than have been declared.

"FILE # NEEDS # RECORDS, HAS ANY #.", implies the file indicated must be reopened to include enough records or another file declared in its place.

ACKNOWLEDGEMENTS:

```
RUN
GET-FILREA
8900 FILES BUS00
RUN
FILREA

HOW MANY FILES ARE AVAILABLE?1


WHEN THE TTY PRINTS A QUESTION MARK (?) TURN ON THE READER
?1
?
 10
?
 2
?
 THIS IS A SAMPLE FILE
?
 1
?
 123
?
 4
?
 2
?
 USED TO DEMONSTRATE
?
 2
?
 FILDUM
?
 2
?
 AND
?
 2
?
 FILREA
?
 1
?
 456
?
 4
?


DONE
```

TITLE:

A REPORT ON FILE CONTENTS AND STRUCTURE

FILRPT
36247

DESCRIPTION:

This program prints a description of each record in a file which includes the amount of numbers and the amount of strings in the record, presence of an end-of-file mark, and the number of words of data in each record and in the entire file.

The program provides a useful check on the contents of a file without the necessity of listing the entire file.

INSTRUCTIONS:

User must input the name of his file as the program requests it.

SPECIAL CONSIDERATIONS:

This program may be adapted to a 2000A, 2000B, or 2000E system by deleting lines 1050-1110 and declaring the file in line 1010.

ACKNOWLEDGEMENTS:

RUN

```
RUN
FILRPT

FILENAME? GENPUR


RECORD 1:  3 STRINGS, 17 WORDS.
RECORD 2:  2 NUMBERS, 4 STRINGS, 39 WORDS.
RECORD 3:  1 STRING, 8 WORDS.
RECORD 4:  0 WORDS.
RECORD 5:  0 WORDS.
RECORD 6:  0 WORDS.
RECORD 7:  0 WORDS.
RECORD 8:  0 WORDS.
RECORD 9:  0 WORDS.
RECORD 10:  0 WORDS.

TOTAL FILE LENGTH = 10 RECORDS, 64 WORDS.
AVERAGE OF 6 WORDS PER RECORD.

DONE


RUN
FILRPT

FILENAME? NAMES


RECORD 1:  3 STRINGS, 59 WORDS.
RECORD 2:  0 WORDS.

TOTAL FILE LENGTH = 2 RECORDS, 59 WORDS.
AVERAGE OF 30 WORDS PER RECORD.

DONE


RUN
FILRPT

FILENAME? IREP


RECORD 1:  16 NUMBERS, 32 WORDS.
RECORD 2:  0 WORDS.
RECORD 3:  3 STRINGS, 10 WORDS.
RECORD 4:  0 WORDS.
RECORD 5:  0 WORDS.
RECORD 6:  0 WORDS.
RECORD 7:  1 NUMBER, 2 WORDS.
RECORD 8:  1 NUMBER, 2 WORDS.
RECORD 9:  1 NUMBER, 2 WORDS.
RECORD 10:  0 WORDS.
RECORD 11:  0 WORDS.
RECORD 12:  0 WORDS.
RECORD 13:  0 WORDS.
RECORD 14:  0 WORDS.
RECORD 15:  0 WORDS.
RECORD 16:  0 WORDS.
RECORD 17:  0 WORDS.
RECORD 18:  0 WORDS.
RECORD 19:  0 WORDS.
RECORD 20:  0 WORDS.

TOTAL FILE LENGTH = 20 RECORDS, 48 WORDS.
AVERAGE OF 2 WORDS PER RECORD.

DONE
```

FINDAD
36867

TITLE:

CONVERTS A FILE TO A FINDIT FILE

DESCRIPTION:

This program copies a source file to a new file which can then be used with the FINDIT Information Retrieval System (HP 36250). The source file may contain both numeric and string data.

INSTRUCTIONS:

The source file must conform to the following conditions:

1. Each logical record must occupy one physical record.
2. Record format must be consistent, each record containing the same number of data items (elements) and in the same order.
3. The first element in each record (ID) must be unique and must begin with a digit.
4. The element on which FINDIT reports will be sorted must be either element 3, 4, 5 or 6 of the source file.
5. Every element must contain data -- null strings are unacceptable.

Before running the program, the FINDIT Master, Aux and Gate files must be opened as specified in the FINDIT manual, HP 36250, Option D00.

The Create program must then be run to initialize the Master file.

The Findad program may then be run. The user responds to the queries as follows:

```
OLD FILE NAME?  (Source file name)
FINDIT FILE NAME?  (Findit Master file name)
PRINTED REPORT (Y/N)? ('Y' prints each record item, 'N' suppresses
    printout)
```

The program always prints each record number and the first data item (ID) of each record as the record is copied. The program prints 'END OF SOURCE FILE' when the source file has been copied.

SPECIAL CONSIDERATIONS:

The user must be familiar with the FINDIT System manual HP 36250, Option D00, especially Sections 7 and 8, Part I.

ACKNOWLEDGEMENTS:

Richard Meyer, HP/Frankfurt
Irv Brenner

```
RUN

OPE-NEWFIL,800
OPE-AUX,32
OPE-GATE,4
OPE-DO,2

GET-CREATE
RUN
CREATE

FILE NAME? NEWFIL
NAME OF AUXILIARY FILE? AUX
NAME OF GATE FILE? GATE
WHAT IS THE SEARCH PASSWORD? SEARCH
WHAT IS THE UPDATE PASSWORD? UPDATE
WHAT IS THE CREATE PASSWORD? CREATE
WHAT IS THE NUMBER (3-6) OF THE ORDERED ELEMENT? 3
TYPE THE ELEMENT NAMES IN THEIR CORRECT SEQUENCE:

 1      ID
 2      FLAG
 3     ?DATE
 4     ?NAME
 5     ?DESCRIPTION
 6     ?XREF
 7     ?XREF2
 8     ?XREF3
 9     ?XREF4
 10    ?

DONE


GET-FINDAD
RUN
FINDAD

CONVERT OLD FILE TO FINDIT FILE.

OLD FILE NAME? TEST
FINDIT FILE NAME? NEWFIL
PRINTED REPORT (Y/N)? Y
 1
ID: 36000A
FLAG: A000
DATE: 00-7XXX-NEW
NAME: NAME
DESCRIPTION: DESCRIPTION
XREF: 123
XREF2: Z99
XREF3: Z99
XREF4: Z99

 2
ID: 36166A
FLAG: A820
DATE: 00-7107-EDU
NAME: DETERM
DESCRIPTION: MATRIX DETERMINANT USING GAUSSIAN ELIMINATION
XREF: 312
XREF2: D15
XREF3: Z99
XREF4: Z99

 3
ID: 36167A
FLAG: A820
DATE: 00-7107-EDU
NAME: JACOBI
DESCRIPTION: EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC MATRIX
XREF: 313
XREF2: 312
XREF3: Z99
XREF4: Z99
```

4
ID: 36116B
FLAG: E211
DATE: 00-7310-REV
NAME: BASTES
DESCRIPTION: BASIC TEST PROGRAM
XREF: B01
XREF2: Z99
XREF3: Z99
XREF4: Z99

5
ID: 36131A
FLAG: B301
DATE: 00-7107-NEW
NAME: CALCOM
DESCRIPTION:
CALCULATOR PROGRAM WITH OPTIONAL PLOTTER OUTPUT (PART 1 OF 2)
XREF: C17
XREF2: K02
XREF3: Z99
XREF4: Z99

6
ID: 36142B
FLAG: A712
DATE: 00-7302-REV
NAME: CSHFL
DESCRIPTION: CASH FLOW ANALYSIS
XREF: 880
XREF2: Z99
XREF3: Z99
XREF4: Z99

7
ID: 36118A
FLAG: A303
DATE: 00-7107-NEW
NAME: CXARTH
DESCRIPTION: VECTOR ARITHMETIC
XREF: Z99
XREF2: Z99
XREF3: Z99
XREF4: Z99

8
ID: 36119A
FLAG: A303
DATE: 00-7107-NEW
NAME: CXEXP
DESCRIPTION: VECTOR EXPONENTIATION
XREF: Z99
XREF2: Z99
XREF3: Z99
XREF4: Z99

9
ID: 36117A
FLAG: B104
DATE: 00-7107-NEW
NAME: DATES
DESCRIPTION: COMPUTES DATE FROM SYSTEM CLOCK
XREF: 108
XREF2: Z99
XREF3: Z99
XREF4: Z99

10
ID: 36144A
FLAG: A302
DATE: 00-7107-NEW
NAME: EXTPRE
DESCRIPTION: 40-DIGIT PRECISION MATHEMATICS
XREF: Z99
XREF2: Z99
XREF3: Z99
XREF4: Z99

11
ID: 36134A
FLAG: A833
DATE: 00-7107-EDU
NAME: H-LIFE
DESCRIPTION: HALF LIFE SIMULATION
XREF: 505
XREF2: 507
XREF3: Z99
XREF4: Z99

12
ID: 36125B
FLAG: B104
DATE: 00-7302-REV
NAME: HELLO
DESCRIPTION: TYPES DATE, TIME, AND PORT NUMBER ON TERMINAL
XREF: 108
XREF2: Z99
XREF3: Z99
XREF4: Z99

13
ID: 36137A
FLAG: A408
DATE: 00-7107-NEW
NAME: KR20
DESCRIPTION: ITEM ANALYSIS AND KUDER-RICHARDSON FORMULA 20 RELIABILITY
XREF: Z99
XREF2: Z99
XREF3: Z99
XREF4: Z99

14
ID: 36145A
FLAG: C107
DATE: 00-7107-NEW
NAME: STGSRT
DESCRIPTION: SORTS STRINGS FROM FILES
XREF: A03
XREF2: S07
XREF3: Z99
XREF4: Z99

15
ID: 36150A
FLAG: A820
DATE: 00-7107-EDU
NAME: NET-3
DESCRIPTION: COMPLEX NUMBER OPERATIONS
XREF: 303
XREF2: 513
XREF3: Z99
XREF4: Z99

CONTRIBUTED PROGRAM **BASIC**

FINDIT
·36250

TITLE:

INFORMATION RETRIEVAL SYSTEM

DESCRIPTION:

The Information Retrieval System is used with the HP 2000C and 2000F. It provides the on-line user with the ability to create, update, and interrogate data files from one or more terminals. Once a file is created, records may be added, modified, and deleted. Any record in the file is available to be printed in a variety of forms such as lists, tabulated reports, or address labels. Records may be retrieved by comparing their contents to a set of file search conditions.

Additional programs permit calculations to be performed on numerical file data, output listings to be ordered on any element, and multi-level file sorts to be performed.

INSTRUCTIONS:

A. File Structure

1. Master file (up to 10,000 records)
2. Auxiliary file (32-record scratch file)
3. Gate file (4-record scratch file)
4. DO file (2-record scratch file)

B. Program Descriptions

1.	CREATE	Used to initially define the master file structure and passwords, and to subsequently modify the file structure.
2.	UPDATE	Used to add, modify, and delete records in the master file.
3.	SEARCH	Retrival program which allows data output in a variety of formats.
4.	FINDIT	Driver program for SEARCH and UPDATE, requiring a password for access to them.
5.	CAL	Calculator program which permits computation and high-precision sums to be made on numerical file data during SEARCH operations.
6.	FIND1	File sort program which permits output listings to be ordered in any element.
7.	IRV1	File sort program which permits multi-level file sorts to be executed.
8.	FINDOR	Driver program for FIND1 and IRV1.

Complete User Instructions are contained in the FINDIT manual HP 36250, option D00.

SPECIAL CONSIDERATIONS:

This system is designed for a data file of 10,000 records or less. Each record may contain approximately 425-500 characters. Element field width is variable. In a typical installation it is recommended that FINDIT be CSAVed and PROtected, except for the program CAL, which should be SAVed and not PROtected.

ACKNOWLEDGEMENTS:

Irv Brenner

```
RUN

SEARCH CONDITIONS:
? STATE=CA
? STATE>R
? STATE=OH
? CLASS<55
? CLASS>62
? cr
  PRINT OPTION? SPECIAL
    ELEMENTS, FIELD WIDTH:
?   ID,4
?   CLASS,5
?   STATE,5
?   NAME,1
?   cr


ID   CLASS STATE NAME

997  43    CA    BELAIR, R. JAMES
2274 51    OH    BURDETT, HENRY
2991 63    TE    BURNS, VIRGINIA
650  24    CA    THOMAS, JOHN F.

TOTAL RECORDS= 4




RUN
FINDIT

FILE: XXXXXX
PASSWORD: XXXXXX
**INVALID PASSWORD.
PASSWORD: XXXXXX
**INVALID PASSWORD.
PASSWORD: XXXXXX
**INVALID PASSWORD.
PASSWORD: XXXXXX
SEARCH CONDITIONS:
? MARITAL=S
?
  PRINT OPTION? SPECIAL
    ELEMENTS, FIELD WIDTHS:
?   NAME,20
?   CITY,10
?   MARITAL,1
?   DEGREE,3
?   CLASS,1
?


NAME                 CITY       M DEG CLASS

ALTMAN, LEO S.       CHICAGO    S BBA 37
BRENNAN,RICHARD R.   WASHINGTON S MBA 53
BURDETT, HENRY       CLEVELAND  S BBA 51
  **2**
BURNS, VIRGINIA      DALLAS     S BJ  63
JACKSON, MILTON      CHICAGO    S BM  49
PRICE, HAROLD        CHICAGO    S BS  49
  **1**
SANTIN, ANDRE        PARIS      S BA  55
TAO, KENNETH         DENNIS     S BA  67
WHITE, SANDRA R.     CLEVELAND  S BFA 61
WILLIS, DONALD L.    WASHINGTON S BA  51
YOUNG, REMINGTON     CHICAGO    S BS  49

TOTAL RECORDS= 11

AGAIN? YES
```

```
SEARCH CONDITIONS:
? CITY=CHICAGO
? MARITAL=S
?
  PRINT OPTION? SPECIAL
    ELEMENTS, FIELD WIDTHS:
?   NAME,20
?   CITY,10
?   DEGREE,3
?   CLASS,1
?


NAME                 CITY       DEG CLASS

ALTMAN, LEO S.       CHICAGO    BBA 37
JACKSON, MILTON      CHICAGO    BM  49
PRICE, HAROLD        CHICAGO    BS  49
YOUNG, REMINGTON     CHICAGO    BS  49

TOTAL RECORDS= 4


AGAIN? YES


SEARCH CONDITIONS:
? STATE=IL
? STATE=OH
? STATE=TE
? MARITAL=S
?
  PRINT OPTION? SPECIAL
    ELEMENTS, FIELD WIDTHS:
?   NAME,20
?   CITY,10
?   STATE,5
?   DEGREE,3
?   CLASS,1
?


NAME                 CITY       STATE DEG CLASS

ALTMAN, LEO S.       CHICAGO    IL    BBA 37
BURDETT, HENRY       CLEVELAND  OH    BBA 51
  **2**
BURNS, VIRGINIA      DALLAS     TE    BJ  63
JACKSON, MILTON      CHICAGO    IL    BM  49
PRICE, HAROLD        CHICAGO    IL    BS  49
WHITE, SANDRA R.     CLEVELAND  OH    BFA 61
YOUNG, REMINGTON     CHICAGO    IL    BS  49

TOTAL RECORDS= 7


AGAIN? NO

DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE: COPIES ONE FILE INTO ANOTHER

FLCOPY
36012

DESCRIPTION:

This program will copy the contents of one file onto another. It handles both string and numeric data as well as end-of-record marks.

INSTRUCTIONS:

Before running the program declare your files in line 8900, old file first then new file. For example:

```
8900 FILES OLDFIL, NEWFIL...
```

Running the program will then copy the contents of the old file into the new file. The contents of the old file will remain unchanged.

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

```
RUN
8900 FILES A
RUN
FILIST

 IS T/S AN HP 2000 'A', 'B', OR 'C'?C

STOP LISTING FILE 1     AT THE FIRST EOF (Y OR N OR Q)?Y

FILE 1     RECORD 1

FOLLOW WINDING PATHS THROUGH THE FORESTS, FOLLOW GENTLE STREAMS
THROUGH LAKES OF BLUE, FOLLOW THE STARS THAT SHINE AT EVEN
WHEN DAY IS THROUGH, WHEN DAY IS THROUGH
DREAM OF THE DAYS THAT PASS BEFORE US
DREAM OF THE INDIAN FIRES GLOWDREAM OF THE LAND WHERE LATIN VOICES
ARE CHANTING LOW, ARE CHANTING LOW
BRING THE WOODED SONGS TO THE CITIES
BRING THE DREAMS OF STARS TO TIRED EYES
BRING HOME THE PATHWASYS OF TOMORROW, FROM THE SKIES,
FROM THE SKIES

FILE 1     RECORD 2

FOLLOW WINDING PATHS THROUGH THE FORESTS, FOLLOW GENTLE STREAMS TO
LAKES OF BLUE, FOLLOW THE STARS THAT SHINE AT EVEN WHEN DAY IS THROUGH
WHEN DAY IS THROUGH            Q

FILE 1     RECORD 3

END OF FILE 1

STOP LISTING FILE 2     AT THE FIRST EOF (Y OR N OR Q)?Q

DONE

GET-FLCOPY
OPEN-FILNEW,10
CRE
8900 FILES A,FILNEW
RUN
FLCOPY
DONE
GET-FILIST
8900 FILES FILNEW
RUN
FILIST

 IS T/S AN HP 2000 'A', 'B', OR 'C'?C

STOP LISTING FILE 1     AT THE FIRST EOF (Y OR N OR Q)?Y


FILE 1     RECORD 1

FOLLOW WINDING PATHS THROUGH THE FORESTS, FOLLOW GENTLE STREAMS
THROUGH LAKES OF BLUE, FOLLOW THE STARS THAT SHINE AT EVEN
WHEN DAY IS THROUGH, WHEN DAY IS THROUGH
DREAM OF THE DAYS THAT PASS BEFORE US
DREAM OF THE INDIAN FIRES GLOWDREAM OF THE LAND WHERE LATIN VOICES
ARE CHANTING LOW, ARE CHANTING LOW
BRING THE WOODED SONGS TO THE CITIES
BRING THE DREAMS OF STARS TO TIRED EYES
BRING HOME THE PATHWASYS OF TOMORROW, FROM THE SKIES,
FROM THE SKIES

FILE 1     RECORD 2

FOLLOW WINDING PATHS THROUGH THE FORESTS, FOLLOW GENTLE STREAMS TO
LAKES OF BLUE, FOLLOW THE STARS THAT SHINE AT EVEN WHEN DAY IS THROUGH
WHEN DAY IS THROUGH            Q

FILE 1     RECORD 3

END OF FILE 1

STOP LISTING FILE 2     AT THE FIRST EOF (Y OR N OR Q)?Q

DONE
```

FMS
36648

TITLE: FILE MANAGEMENT SYSTEM

DESCRIPTION:

This program is a series of routines that may be used in manipulating files.

INSTRUCTIONS:

A FILES statement is to be entered in statement 1. The routines are accessed by the commands:

COMP This routine compares selected records of file number 1 with selected records of file number 2. The routine requests the beginning and ending records of file number 1 and the beginning record of file number 2. Records that are different are listed on the teletype.

COPY This routine copies file number 1 into file number 2.

DUMP This dumps the contents of files number 1 up to 4 on paper tape. For the dump to be accurate a certain restriction is necessary on the allowed characters in the strings stored in the file. The restrictions are:

No V^C as the first character,

No U^C as the first character,

No Q^C as the first character,

No $\uparrow^C$ anywhere, and

No $\leftarrow^C$ anywhere.

HELP Gives a list and short description of the commands.

LIST This lists any file on the teletype.

READ This is a routine to restore files from a paper tape dumped by the DUMP command.

STOP Halts the program.

SUMR This program summarizes the structure of any file. One line of either "N" or "S" is typed out (with a space every 10 characters) to indicate a number or a string respectively at that particular location in a record. Thus one can determine what type of data items are present without listing the entire file.

TRAN Transfers selected records from file number 1 to file number 2. The routine requests beginning and ending records of file number 1 and the beginning record of file number 2.

SPECIAL CONSIDERATIONS:

The use of control characters is restriced for the DUMP routine. The resultant paper tape dump is about one-half that of FILDUM (HP 36008).

In dumping a file containing floating point number, remember that BASIC rounds numbers when they are printed. Thus a restored file may be different in the last decimal place on some numbers.

ACKNOWLEDGEMENTS: Lawrence E. Turner, Jr.
Pacific Union College

```
RUN

1  FILES A,B
RUN
FMS


COM ?HELP


                FILE MANAGEMENT SYSTEM

THIS PROGRAM PROVIDES ROUTINES FOR FILE MANIPULATION.
THESE ARE ACCESSED BY THE FOLLOWING COMMANDS:

COMP:  COMPARES FILE #1 WITH FILE #2, BEGINNING AND ENDING
       RECORDS ARE REQUESTED.

COPY:  COPIES FILE #1 INTO FILE #2.

DUMP:  DUMPS THE CONTENTS OF UP TO 4 FILES ONTO PAPER TAPE.

HELP:  GIVES THE COMMANDS.

LIST:  LISTS THE CONTENTS OF A FILE ON THE TTY.

READ:  RESTORES FILES FROM PAPER TAPE PREVIOUSLY DUMPED BY
       THE 'DUMP' COMMAND.

STOP:  HALT EXECUTION.

SUMR:  SUMMARIZES THE STRUCTURE OF A FILE.

TRAN:  TRANSFERS SELECTED RECORDS FROM FILE #1 TO FILE #2.
       ZERO ENTRIES FOR RECORD NUMBERS RETURNS TO COMMAND.


COM ?LIS

WHICH FILE TO BE LISTED ?1
ENTER: BEG & END RECORDS ?1,6
STOP AT FIRST EOF ?N


    ********* BEGIN *********

7856.45
12

 1     ***** EOR; 7    WORDS, 2     ITEMS *****

THIS IS THE SECOND STRING

 2     ***** EOR; 14   WORDS, 1     ITEMS *****

MAY, 1973
THIRD STRING

 3     ***** EOR; 13   WORDS, 2     ITEMS *****

THE LIVING END

 4     ***** EOR; 8    WORDS, 1     ITEMS *****

 5     ***** EOF; 0    WORDS, 0     ITEMS *****

 6     ***** EOF; 0    WORDS, 0     ITEMS *****



COM ?COPY

COPY COMPLETED, 48   RECORDS.
```

```
COM ?TRAN

ENTER:  B1, E1, B2 ?3,3,4

ENTER:  B1, E1, B2 ?0,0,0


COM ?COMP

ENTER:  B1, E1, B2 ?1,6,1

COMPARE OF RECORDS 1    AND 1    GOOD, 2    ITEMS
COMPARE OF RECORDS 2    AND 2    GOOD, 1    ITEMS
COMPARE OF RECORDS 3    AND 3    GOOD, 2    ITEMS
COMPARE OF RECORDS 4    AND 4    FAILED AT ITEM 1

**** FILE 1    RECORD 4    ****

THE LIVING END
**** EOR
**** FILE 2    RECORD 4    ****

MAY, 1973
THIRD STRING
**** EOR
COMPARE OF RECORDS 5    AND 5    GOOD, 0    ITEMS
COMPARE OF RECORDS 6    AND 6    GOOD, 0    ITEMS


COM ?SUMR

WHICH FILE TO BE SUMMARIZED ?1

 SS

 1    EOR;  7    WORDS, 2    ITEMS.

 S

 2    EOR;  14   WORDS, 1    ITEMS.

 SS

 3    EOR;  13   WORDS, 2    ITEMS.

 S

 4    EOR;  8    WORDS, 1    ITEMS.


 5    EOF;  0    WORDS, 0    ITEMS.


 6    EOF;  0    WORDS, 0    ITEMS.


 7    EOF;  0    WORDS, 0    ITEMS.


 8    EOF;  0    WORDS, 0    ITEMS.


 9    EOF;  0    WORDS, 0    ITEMS.


 10   EOF;  0    WORDS, 0    ITEMS.
```

•
•
•

```
COM ?DUMP

HOW MANY FILES TO BE DUMPED ?1
STOP DUMP AT FIRST EOF ?YES


TURN ON TAPE PUNCH (LEADER OK?).
 1
 48
7856.45
12
EOR
THIS IS THE SECOND STRING
EOR
MAY, 1973
THIRD STRING
EOR
THE LIVING END
EOR
EOI


END OF DUMP, PUNCH OFF.
COM ?STOP


DONE
```

```
*************************************************************
*                                                           *
*  1.  KEYWORDS - PARAMETERS NOT ENCLOSED IN <>             *
*                                                           *
*  2.  []       - ANY PARAMETERS ENCLOSED IN BRACKETS ARE   *
*                 OPTIONAL.                                 *
*                                                           *
*  3.  ()       - WHEN TWO OR MORE PARAMETERS APPEAR        *
*                 WITHIN PARENTHESES, AT LEAST ONE OF       *
*                 THE PARAMETERS MUST BE GIVEN.             *
*                                                           *
*  4.  /        - 'OR'                                      *
*                                                           *
*  5.  <CHAR>   - ANY CHARACTER EXCEPT "," OR ";"           *
*                                                           *
*  6.  <NUM>    - ANY NON-NEGATIVE UNSIGNED INTEGER         *
*                                                           *
*  7.  <STRING> - <DELIMITER><CHARSTRING><DELIMITER>        *
*                                                           *
*  8.  <CHARSTRING> - <CHAR>[<CHARSTRING>]                  *
*                                                           *
*  9.  <DELIMITER> - <CHAR> NOT APPEARING IN <CHARSTRING>   *
*                                                           *
*************************************************************
```

TABLE 1-1
COMMAND DEFINITION NOTATION

5. PAPER
6. TOPSPACE

B. TEXT INPUT CONTROL

TEXT INPUT TO THE FORMATTER CAN BE PROVIDED FROM THREE DIFFERENT SOURCES: A TEXT FILE, AN AUXILIARY HOLD FILE OR THE KEYBOARD. THE COMMANDS AVAILABLE ARE:

1. TEXT
2. HOLD
3. ENTER

C. PAGE FORMAT CONTROL

THE FOLLOWING COMMANDS ALLOW THE USER TO CONTROL THE CONTENT ON EACH PAGE AND THE SPACING BETWEEN THE LINES.

1. NEED
2. NEWPAGE
3. ODDPAGE
4. SKIP
5. SPACING

D. LINE FORMAT CONTROL

LINE FORMAT CONTROL COMMANDS ALLOW THE USER CONTROL OVER HOW THE OUTPUT TEXT WILL APPEAR.

1. ADJUST
2. BLANK
3. BREAK
4. CENTER
5. FILL
6. INDENT
7. SUPPRESS
8. TAB
9. UNDENT

E. MISCELLANEOUS

THE FOLLOWING COMMANDS PROVIDE ADDITIONAL FACILITIES TO THE USER.

1. CHECK
2. CONTROL
3. DEFINE
4. FLAG
5. OPEN
6. PAUSE
7. REPEAT

2.0 OPERATING PROCEDURES

2.1 FORMATTER INITIALIZATION

A USER LOGGED ONTO A 2000C HIGH SPEED OR 2000F TIME SHARE SYSTEM MAY OBTAIN A COPY OF THE FORMATTER BY ENTERING THE COMMAND:

GET-FORM2K

PRIOR TO EXECUTING THE PROGRAM, ANY DESIRED UTILITY FILES (COMMAND FILE OR NEED FILE) SHOULD BE CREATED VIA THE TSB SYSTEM 'OPEN' COMMAND IF THEY DO NOT ALREADY EXIST. EXECUTION IS THEN INITIATED BY ENTERING THE COMMAND:

RUN

THE PROGRAM RESPONDS WITH THE CONTROL CHARACTER AS A PROMPT. IF THE USER WISHES TO MAKE USE OF ANY UTILITY FILES, THEY SHOULD BE OPENED WITH THE FORMATTER 'OPEN' COMMAND AT THIS TIME. FOR A DESCRIPTION OF HOW TO SPECIFY A UTILITY FILE, SEE THE 'OPEN' COMMAND.

THE USER MAY THEN ENTER ANY DESIRED INITIALIZATION COMMANDS NECESSARY TO TAILOR THE FORMATTER TO THE IMMEDIATE TASK. REFER TO APPENDIX A FOR THE FORMATTER PARAMETER DEFAULT VALUES AND WHICH COMMANDS CAN BE EXECUTED DURING INITIALIZATION. NOTE THAT AT LEAST ONE 'TEXT' COMMAND MUST BE PROVIDED. EACH TIME A STRING OF FORMATTER COMMANDS IS EXECUTED, THE PROGRAM RETURNS TO THE USER WITH THE PROMPT CHARACTER TO REQUEST MORE INPUT.

IF THE INITIAL VALUES (SEE APPENDIX A) ARE SATISFACTORY, THE USER NEED ONLY ENTER THE TEXT FILE NAME IN THE FORM:

TEXT <FNAME>

A CARRIAGE RETURN IN RESPONSE TO THE CONTROL CHARACTER PROMPT WILL TERMINATE INITIALIZATION AND START THE FORMATTING PROCESS.

2.2 PERFORMANCE INPROVEMENT SUGGESTIONS

INPUT COMMANDS AND PARAMETERS MAY BE EITHER UPPER OR LOWER CASE FOR THE CONVIENCE OF THE USER. THE INITIAL COMMAND SCAN, HOWEVER, WILL BE FOR UPPER CASE CHARACTERS. IF A COMMAND IS NOT IDENTIFIED, THE PROCESS IS REPEATED FOR THE LOWER CASE. PERFORMANCE CAN THUS BE IMPROVED BY USING UPPER CASE LETTERS EXCLUSIVELY.

CONTRIBUTED PROGRAM **BASIC**

FORMAT
36005

TITLE:

ALLOWS SPECIAL FORMATTING OF DATA PRINTOUT

DESCRIPTION:

The format subroutine allows the user to exercise greater latitude in specifying the printing format. This permits dynamic changes in the printing format during execution.

INSTRUCTIONS:

Using the stringZ$, the format is specified by:

1. A decimal point (.) in the position the decimal point is desired.
2. The number sign (#) in each position to be reserved for printing a digit. Note leading zeros are suppressed; trailing zeros are printed.
3. A plus sign (+) or space preceding the first digit position if it is desired to have the polarity printed.

String characters other than those mentioned will be printed out as designated. The string Z$ is dimensioned in the subroutine. Variables to be printed are transferred to the subroutine by means of Z(1),Z(2).... Where Z(1) is the Ith number to be printed. If more than 10 numbers are to be printed, the Zarray must be dimensioned accordingly.

SPECIAL CONSIDERATIONS:

If the variable given is larger than the format space provided, the subroutine will print #'s in place of the value. Since the system resolution is 6+ digits, the use of more than 6 #'s for a single variable can result in the printing of arbitrary digits.

ACKNOWLEDGEMENTS: Hobbs Associates

RUN

```
GET-DEMO
LIS
DEMO

10  REM PROGRAM TO DEMONSTRATE FORMAT
20  LET Z[1]=1
30  LET Z[2]=-1
40  LET Z[3]=.00432
50  LET Z[4]=Z[5]=-123.456
60  LET Z$="ZEROES ##.## ,SIGN +#.# ,DECIMAL .#####, ###.# +###.###"
70  GOSUB 9003
80  STOP


APPEND-FORMAT
RUN
DEMO

ZEROES  1.00 ,SIGN -1.0 ,DECIMAL .00432, 123.5 -123.456
DONE
```

TITLE:

F AND I FORMAT

FORMIF
36612

DESCRIPTION:

This is a subroutine to print numbers in FORTRAN style formats.
(Iw and Fw.d):

INSTRUCTIONS:

Input values: Z, value of quantity to be printed;
W9, field width;
D9, number of decimal places.

The field begins wherever the carriage is located and the carriage is left at the end of the field after the quantity is printed.

Entry points: 9910 integer format
9920 floating format

SPECIAL CONSIDERATIONS:

The variables used are: Z$(10),Z0,Z1,Z2,Z3,Z7,Z8,Z9.
The length is 450 words which may be reduced to 323 words by deleting REM's.

ACKNOWLEDGEMENTS:

Lawrence E. Turner
Pacific Union College

```
RUN

NAM-TEST
10   INPUT Z
20   PRINT "Z= ";
30   LET W9=9
40   LET D9=3
50   GOSUB 9920
60   PRINT
70   PRINT
80   GOTO 10

APP-FORMIF
9999 END

RUN
TEST

?1.234567
Z=      1.235

?103.0
Z=    103.000

?.001456732
Z=      0.001

?.0015123
Z=      0.002

?1700
Z=   1700.000

?
DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE:

FUNCTION PLOT

FPLOT
36112

DESCRIPTION:

This program plots any function that can be expressed as Y = F(X) on the Model 7200A Graphic Plotter. Up to 10 other parameters can be used and changed on succeeding plots. For example, the value of N in $Y = X^N$ can be changed for each plot by inputting a value for the parameter N. It also draws axes and scales automatically if desired.

INSTRUCTIONS:

All instructions are included in the program.

INPUTS (Automatic Scaling)	FORMAT
1. Instructions	Yes or No (Answer "Yes"
2. X Min.?	the first time you RUN)
3. X Max.?	
4. Delta X?	
5. Format: A = Automatic Scaling B = Input Graph Limits	A or B
6. Do you have parameters to input?	Yes or No
7. How many?	1 to 10
8. Grid Size In Major Divisions Width?	
9. Height?	
10. Axes?	Yes or No
11. N = ?	
12. Another Graph On Same Scale?	Yes or No
13. Input New Parameter Values: N = ?	

INPUTS (Non-Automatic Scaling)

Delete: 8 and 9 above
Add: Enter Values for Edges of Graph: Left? Bottom?
Right? Top?

SPECIAL CONSIDERATIONS:

Enter your function expressed as Y = f(x) beginning on line 100.
For example: 100 Y = X↑3. Type RUN.

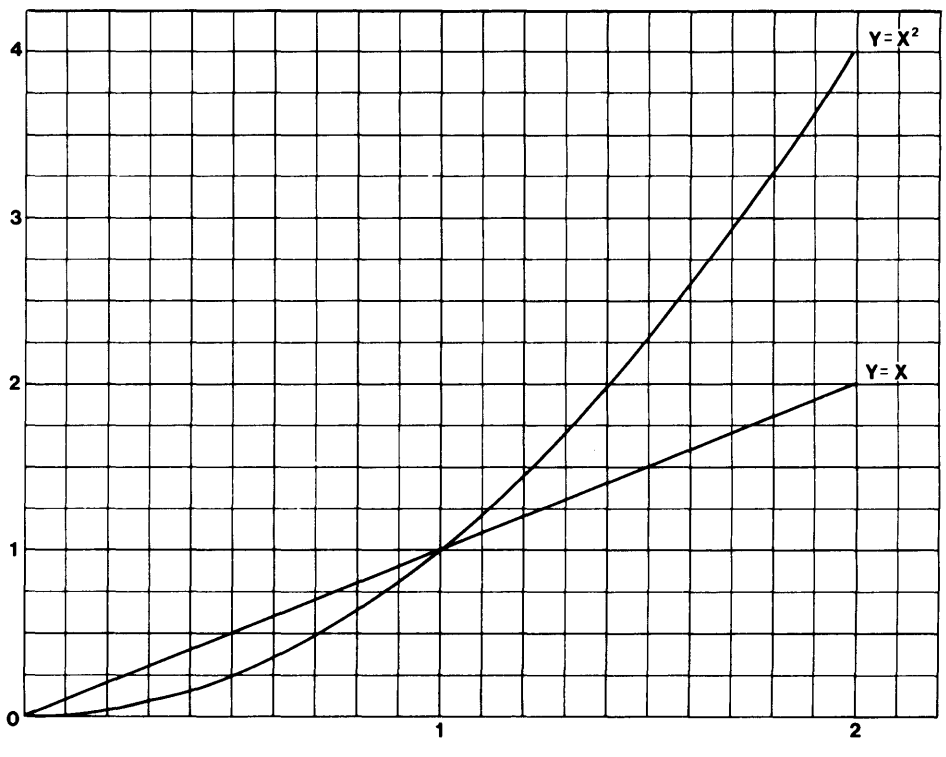
ACKNOWLEDGEMENTS:

Hewlett-Packard/San Diego

RUN

```
RUN
FPLOT

INSTRUCTIONS?YES
ENTER YOUR FUNCTION EXPRESSED AS Y=F(X) BEGINNING
ON LINE 100.  FOR EXAMPLE:

100 Y=X↑3

YOU WILL THEN BE ASKED QUESTIONS ABOUT THE WAY YOU
WISH THE PLOT TO BE PRESENTED:

1. X MIN AND X MAX: INPUT THE SMALLEST AND LARGEST
VALUES FOR X YOU WISH PLOTTED.

2. DELTA X: INPUT THE INCREMENT OF X BETWEEN CALCULATED
DATA POINTS.  USUALLY A VALUE 1/100 THE DIFFERENCE
BETWEEN X MAX AND X MIN IS SUFFICIENT.

3. FORMAT: ASKS WHETHER YOU WISH AUTOMATIC SCALING OR
A FIXED SCALE OF YOUR CHOICE.  IF THE FIXED SCALE IS
CHOSEN YOU WILL BE ASKED FOR THE VALUES OF X AND Y
AT THE EDGES OF THE GRAPH.

4. PARAMETERS: ANSWER YES OR NO, WHETHER YOU HAVE
A PARAMETER OTHER THAN X YOU WISH TO INPUT DURING
PROGRAM EXECUTION (MORE ON THIS LATER.)

5. AXES: ANSWER YES OR NO,WHETHER YOU WISH AXES DRAWN
ON THE GRAPH.  TIC MARKS WILL BE DRAWN IF AN AXIS IS
OFF SCALE.

YOU MAY INCLUDE OTHER PARAMETERS TO INPUT, FOR EXAMPLE:

100 Y=X↑N

THE VARIABLE N MUST BE USED, AND THE COMPUTER WILL
ASK N=?  IN THE PROGRAM.  IF MORE THAN ONE PARAMETER
IS NEEDED, LABEL THEM N(1) UP TO N(10).  EXAMPLE:

100 Y=N(1)*X↑2+N(2)*X+N(3)

IS AN EXPRESSION USABLE FOR ANY QUADRATIC EQUATION.


NOW ENTER YOUR FUNCTION, FOLLOWED BY 'RUN'.

DONE
100 Y=X↑N
RUN
FPLOT

INSTRUCTIONS?NO
X MIN ?0
X MAX ?2
DELTA X?.1
FORMAT:A=AUTOMATIC SCALING,B=INPUT GRAPH LIMITS?A
DO YOU HAVE PARAMETERS TO INPUT?YES
HOW MANY?1
GRID SIZE IN MAJOR DIVISIONS
WIDTH?15
HEIGHT?10
AXES?NO
ENTER INPUT PARAMETER(S):
N=?1\

??2
SCALING
X AXIS FROM   0    TO   3
Y AXIS FROM   0    TO   5
PLTL
0000 0000↑
0333 0019
0666 0079
0999 0179
1333 0319
```

```
1666 0499
1999 0719
2333 0979
2666 1279
2999 1619
3333 1999
3666 2419
3999 2879
4332 3379
4666 3919
4999 4499
5332 5119
5666 5779
5999 6479
6332 7219
6665 7999
6666 7999
      PLTT
ANOTHER GRAPH ON SAME SCALE?YES
INPUT NEW PARAMETER VALUES:
N=?1
PLTL
0000 0000↑
0333 0199
0666 0399
0999 0599
1333 0799
1666 0999
1999 1199
2333 1399
2666 1599
2999 1799
3333 1999
3666 2199
3999 2399
4332 2599
4666 2799
4999 2999
5332 3199
5666 3399
5999 3599
6332 3799
6665 3999
6666 3999
      PLTT
ANOTHER GRAPH ON SAME SCALE?NO

DONE
```



RUN

```
GET-FPLOT
RUN
FPLOT

INSTRUCTIONS?NO
PLACE YOUR FUNCTION EXPRESSED AS Y=F(X) ON LINES
100 TO 500 AND THEN RUN AGAIN.

DONE
100 Y=X↑N
RUN
FPLOT

INSTRUCTIONS?NO
X MIN ?0
X MAX ?2
DELTA X?.1
FORMAT:A=AUTOMATIC SCALING,B=INPUT GRAPH LIMITS?B
DO YOU HAVE PARAMETERS TO INPUT?YES
HOW MANY?1
ENTER VALUES FOR EDGES OF GRAPH:
LEFT?0
RIGHT?3
BOTTOM?0
TOP?5
AXES?NO
ENTER INPUT PARAMETER(S):
N=?1


SCALING
X AXIS FROM   0    TO   3
Y AXIS FROM   0    TO   5
PLTL
ANOTHER GRAPH ON SAME SCALE?YES
INPUT NEW PARAMETER VALUES:
N=?2
PLTL
ANOTHER GRAPH ON SAME SCALE?NO

DONE
```

CONTRIBUTED PROGRAM **BASIC**

GRAPHS
36115

TITLE: DEMO PROGRAM FOR MODEL 7200A GRAPHIC PLOTTER

DESCRIPTION:

GRAPHS provides line graphs, step graphs, point or bar graphs from a data file.

INSTRUCTIONS:

Instructions are self-explanatory.

Two tapes are included in this package
1. Type "~~OPEN~~-GRAP1, 2" ~~Then~~ enter Entry Tape for GRAP1 and RUN.
2. Enter GRAPHS Tape

INPUTS	FORMAT
Want Listing Of Actual Data Points?	Yes or No.
Type Of Graph:	1 or 2 or 3 or 4
Another Graph	Yes or No

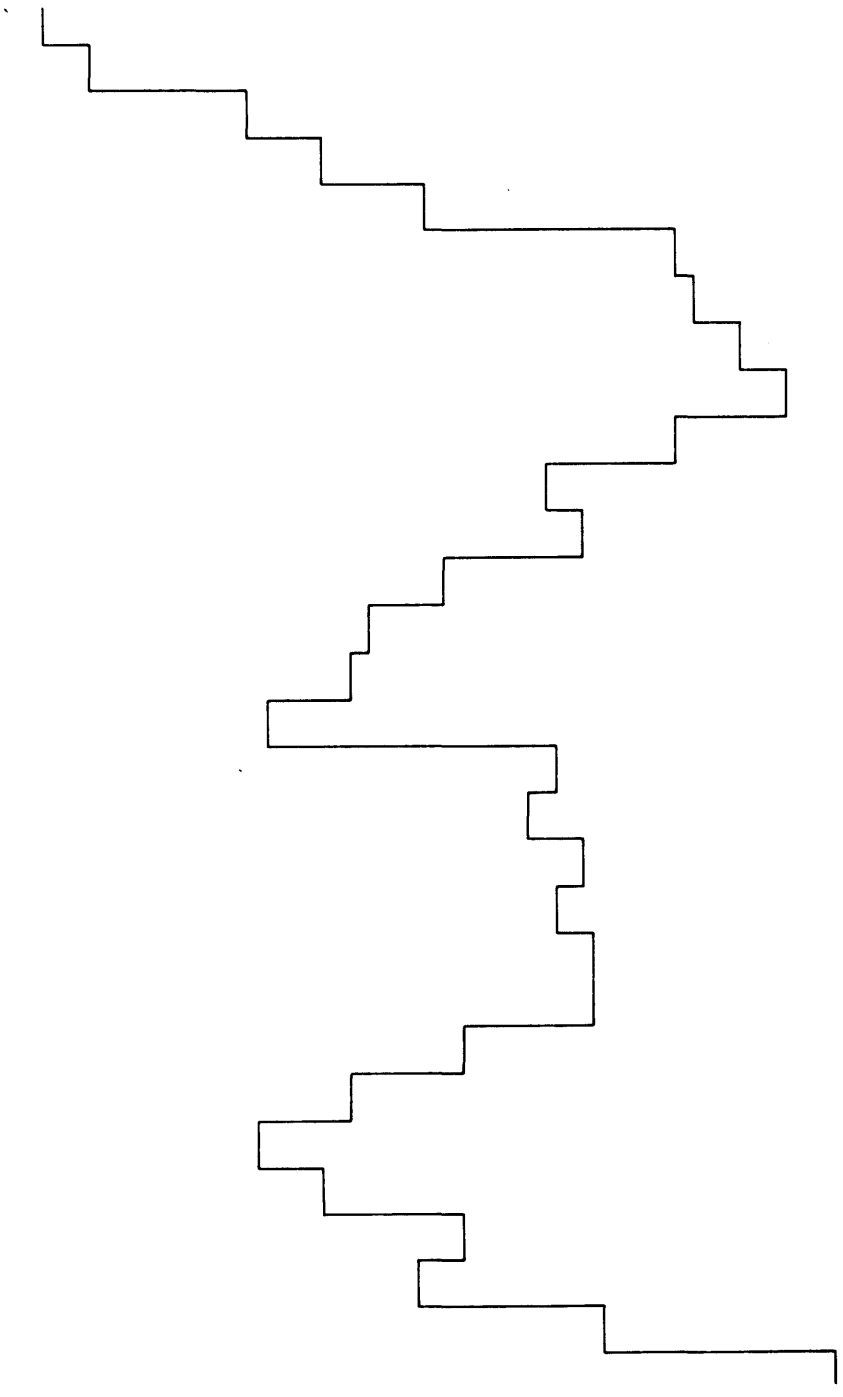
FILES

GRAP1, 2

ACKNOWLEDGEMENTS: Hewlett-Packard/San Diego

RUN

```
RUN
GRAPHS

WANT LISTING OF ACTUAL DATA POINTS?NO
TYPE OF GRAPH: LINE = 1
               BAR  = 2
               STEP = 3
               POINT= 4
?3
PLTL
ANOTHER GRAPH?NO

DONE
```



CONTRIBUTED PROGRAM **BASIC**

GTAPID 36548

TITLE: PAPER TAPE TITLER

DESCRIPTION:

This program allows the user to punch a title on paper tape in readable form -- i.e., the holes punched will form characters that may be read visually.

INSTRUCTIONS:

When run, the program asks for a title. After it has been typed, but before hitting the carriage return, turn on the paper tape punch. The desired title will be punched in visual form on the tape.

The program operates by printing characters that have punched representations which will collectively form the desired characters. The results produced on the printed output will bear no particular relationship to the desired title and should be ignored.

SPECIAL CONSIDERATIONS:

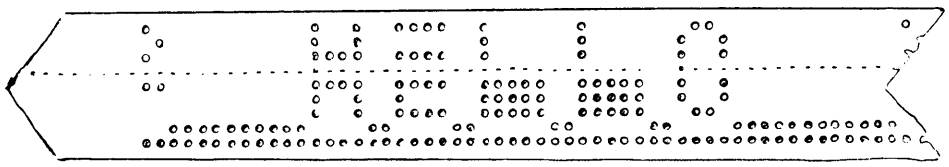
None

ACKNOWLEDGEMENTS:

Graduate School of Business
Stanford University

RUN

```
RUN
GTAPID

TITLE -- ?HELLO
●●●●●●●●●●??●●?---●●?888888●?8888●●!!●●●●●●●●●●●●
DONE
```



CONTRIBUTED PROGRAM **BASIC**

TITLE: HAZELTINE 2000 USER SUBROUTINES

HAZEL
36786

DESCRIPTION:

These two subroutines allow one to use the HAZELTINE 2000 terminal effectively for graphics and other control processes.

The subroutine FAZEL uses the file $CHARS (HP 36220) for the HP 2000F or $CHARSE (HP 36757) for the HP 2000E.

INSTRUCTIONS:

This describes the hardware and software characteristics necessary for using the HAZELTINE 2000 computer terminal.

Besides behaving as a normal (but high speed) teletype, it has certain graphic and control capabilities.

"BASIC" does not recognize certain characters, but the file $CHARS (or $CHARSE) may be used to generate these. This accounts for the existence of two programs with essentially the same purpose.

The HAZELTINE is different from a teletype in that the LF function is ignored. An automatic linefeed is provided with each CR by the HAZELTINE.

NAME OF SYSTEM HAZEL

MAIN PROGRAM HAZEL

NAMES OF ALL SUPPORTING PROGRAMS: FAZEL

Continued on following page.

SPECIAL CONSIDERATIONS:

Note: When using the subroutine FAZEL on an HP 2000E, change line 9505 to read:

```
9505  FILES $CHARSE
```

ACKNOWLEDGEMENTS: Ted Park
Pacific Union College

INSTRUCTIONS continued

SOFTWARE

The programs HAZEL and FAZEL are essentially identical. Only a certain number of special characters are allowed in string variables by BASIC. Therefore, program HAZEL is only able to address the screen in a rather hit-and-miss manner (see below for dis-allowed values). Program FAZEL uses the special file $CHARS (or $CHARSE for the HP 2000E) which contains the complete ASCII character set. Therefore, FAZEL is able to address the entire screen. FAZEL has the disadvantage, however, of using a file.

All of the subroutines at the right use no internal variables except for subroutine 9500 which uses:
Z Z0 Z1 X$ Y$

9000	CLEAR SCREEN
9100	CLEAR BOLDFACE
9200	HOME CURSOR
9300	LIGHTFACE
9400	BOLDFACE
9500	ADDRESS CURSOR
9600	EMPTY BASIC LINE BUFFER

Subroutine 9500 must be given X and Y coordinates in the variables X and Y which are not changed by the subroutine. In each case: 0<=X<=73 and 0<=Y<=26. Values which are either too large or too small are positioned at the nearest boundary.

Subroutine 9500 returns the value - Z with the following meaning:

0 coordinates OK
1 X coordinate illegal (for HAZEL only)

ADDITIONAL INFO FOR HAZEL	ADDITIONAL INFO FOR FAZEL
Subroutine 9500 uses the variables: Z2 D$	Subroutine has a FILES statement in line 9505: 9505 FILES $CHARS (Change to $CHARSE for use on the HP 2000E)
Illegal X values are: 0,10,13,14,15,19,24,34	This subroutine has no illegal X values.

NOTE: DO NOT POSITION THE CURSOR AT THE BOTTOM RIGHT-HAND CORNER (73,26) AS THIS WILL CAUSE AN AUTOMATIC "ROLL-UP" OF THE PREVIOUSLY DRAWN MATERIAL.

INSTRUCTIONS continued

HARDWARE

$>^C$	lead-in
N^C	transmit
Q^C	address cursor*
R^C	home cursor
S^C	delete line
Y^C	lightface
Z^C	insert line
$\backslash^C$	clear screen
$]^C$	clear boldface
$\uparrow^C$	print
$\leftarrow^C$	boldface

* must be followed by X and Y position selected from table at right:
0<=X<=73
0<=Y<=26.
(the Y position may be used without control: i.e. A does the same as A^C.)

BASIC does not allow:
$@^C$ J^C M^C N^C O^C S^C X^C "

EXAMPLES:

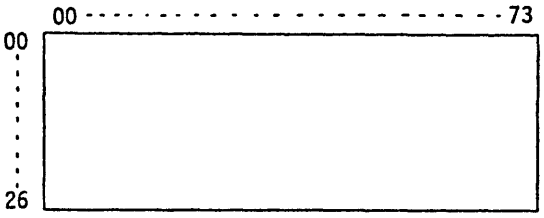
1. to change to boldface:
   type $>^C$ $\leftarrow^C$

2. to delete line:
   type $>^C$ S^C

3. to position cursor at 5^{th} row and 12^{th} column:
   type $>^C$ Q^C L^C E^C

00	$@^C$	27	$]^C$	48	0
01	A^C	28	$\backslash^C$	49	1
02	B^C	29	$[^C$	50	2
03	C^C	30	$\uparrow^C$	51	3
04	D^C	31	$\leftarrow^C$	52	4
05	E^C	32	space	53	5
06	F^C	33	!	54	6
07	G^C	34	"	55	7
08	H^C	35	#	56	8
09	I^C	36	$	57	9
10	J^C	37	%	58	:
11	K^C	38	&	59	;
12	L^C	39	'	60	<
13	M^C	40	(	61	=
14	N^C	41	)	62	>
15	O^C	42	*	63	?
16	P^C	43	+	64	@
17	Q^C	44	,	65	A
18	R^C	45	-	66	B
19	S^C	46	.	67	C
20	T^C	47	/	68	D
21	U^C			69	E
22	V^C			70	F
23	W^C			71	G
24	X^C			72	H
25	Y^C			73	I
26	Z^C				

ROWS AND COLUMNS ON HAZELTINE CRT



(the rows are 6/inch and the columns are 10/inch so that even though the screen is almost square, the numbering is as above.)

CONTRIBUTED PROGRAM **BASIC**

HELLO
36125

TITLE:

TYPES DATE, TIME, AND PORT NUMBER ON TERMINAL

DESCRIPTION:

The program accesses the 2000B/C Timeshare System Clock; formats the date, time, and port numbers into string format; prints it out onto the user's terminal when he logs on to the system.

Additional messages may be printed by inserting PRINT instructions as indicated in the program.

The program automatically adjusts for leap year.

Annually, at year's end the program automatically wraps around to January 1 for the new year.

INSTRUCTIONS:

Identify the system ownership in line 190.

Add additional print statements as needed between lines 160 and 170.

SPECIAL CONSIDERATIONS:

A new system date must be entered at "Awake" time at least once every 999 days.

ACKNOWLEDGEMENTS:

Peter Wolmut and James R. Hansz
Multnomah County IED

```
RUN

RUN
HELLO

01-10-73 09:54 AM  PORT #28

TEST OF LIBRARY PROGRAM

DONE
```

CONTRIBUTED PROGRAM **BASIC**

HISS
36235

TITLE:

SAMPLE STATISTICS AND HISTOGRAM FORMED FROM A SET OF NUMBERS

DESCRIPTION:

This program provides sample statistics and produces a histogram from a set of data. As normally run, an automatically scaled histogram of each data point is produced. Special options provide the capability of tailoring the histogram to the user's needs.

INSTRUCTIONS:

Data to be used for the histogram should be entered as data statements on lines 20 through 999. When the RUN command is given, a histogram will be produced in accordance with certain predetermined options. The DATA lines containing these options and their effect on the histogram are described below.

Line 11 TITLE OPTION (Normally zero)

If the zero is changed to any positive number, a title of up to 72 characters enclosed in quotes may be inserted immediately following the number. The title will be printed centered on the page.

(continued page 2)

SPECIAL CONSIDERATIONS:

1. Maximum number of data points is 1000.
2. Largest data point must be less than 1.E+35.

ACKNOWLEDGEMENTS:

R. G. Hewitt
Ampex Corporation

INSTRUCTIONS (Cont'd.)

Line 12 LIMIT OPTION (Normally zero)

For certain types of data, it is often desirable to graphically display the limits between which the data is normally expected to fall. This may be done by changing the zero to any positive number, followed by the lower and upper limits. A series of dashes will appear across the page when the number entered falls within one of the cells on the histogram. If there is only one limit (lower or upper) the unused limit should be entered as "1.E+34".

Line 13 Y AXIS OPTION (Normally zero)

With zero entered, the cell size is determined to be the lowest difference between any two data points. This data line may be changed to start with any number between 1 through 8, immediately followed by the parameters required in accordance with the table below.

With one entered:	The cell size must be specified
With two entered:	Specify lowest cell number and cell size
With three entered:	Specify lowest cell number and number of cells
With four entered:	Specify lowest cell, number of cells, and highest cell
With five entered:	Specify lowest cell, cell size and highest cell
With six entered:	Specify highest cell and number of cells
With seven entered:	Specify highest cell and cell size
With eight entered:	Specify highest cell

NOTE: Cell size is limited to three significant digits.

Line 14 CELL DEFINITION (Normally zero)

With zero entered data points within 1/2 cell number above and within 1.E-34 of 1/2 cell number below the cell number are included in the cell.

With one entered: Data points between the cell number and the next lowest cell number are included in the cell.

With two entered: Data points between the cell number and the next highest cell number are included in the cell.

Line 15 X AXIS OPTIONS (Normally One)

With zero entered the carriage is advanced one space for each data point within the cell. If there are more than 60 data points within the cell, control is automatically transferred as described in one below.

With one entered the amount the carriage is advanced is determined automatically by the cell with the highest number of data points in accordance with the following table. Full scale equals 60 spaces.

Per Data Point		Max. Data Points
5		12
2		30
1		60
.5	Rounded to closest integer	120
.1		600
.05		1200

With two entered, the data points per cell are plotted as their percent of the total.

With three entered the data points per cell are plotted as the percent of the cell (or cells) containing the largest number of data points.

Line 16 PRINT LINES PER CELL (Normally one)

The number entered determines the number of print lines per cell. Number's 1, 3 and 5 are the valid entries.

Line 17 CELL LABEL (Normally one)

The number entered determines the interval of cell labels, i.e., 2 will result in every other cell being labeled, 3 will result in every third cell, etc. The first and last cell will always be labeled. If limits have been entered in line 2 they will also be labeled.

SPECIAL INSTRUCTIONS

FILES:

The program can be run from data files. In order to do so, enter at line 1005 "FILES (NAME OF FILE)", and change line 1710 to "READ#1;A(J)". The last data item in the file should be greater than 1E+34.

SORTED DATA

The program sorts the data in ascending order. If there is a large number of data points, this can take a considerable length of time. If the data has been previously arranged in ascending order, the program efficiency can be improved by deleting lines 1780 through 1860.

MULTIPLE RUNS WITH SAME DATA

In order to optimize the appearance of the histogram, it is sometimes desirable to make several runs with the same data while changing the options provided in lines 11 through 17. If there are large numbers of data points, program efficiency can be improved as follows:

1. Open a file of suitable size to contain all the data points.
2. Enter the data in lines 20 through 999.
3. Add the following lines to the program:

```
1005    FILES  (name of file)

1855    PRINT #1; A (K)

1865    PRINT #1; A (K)
```

4. Run the program. This produces a file of data that is sorted in ascending order.
5. Delete lines 1855 and 1865.
6. Delete lines 1780 through 1860 as per "SORTED DATA".
7. Change line 1710 to "READ #1; A(J) as per "FILES".
8. Make the desired changes in lines 11 through 17 and rerun the program.

```
GET-HISS
TAPE
20 DATA 45.405
21 DATA 45.4,45.39
22 DATA 45.4,45.39,45.38
23 DATA 45.4,45.39,45.38,45.37
24 DATA 45.4,45.39,45.38,45.37,45.36
25 DATA 45.4,45.39,45.38,45.37,45.36,45.35
26 DATA 45.4,45.41,45.42,45.43,45.44,45.45
27 DATA 45.4,45.41,45.42,45.43,45.44
28 DATA 45.4,45.41,45.42,45.43
29 DATA 45.4,45.41,45.42
30 DATA 45.4,45.41
31 DATA 45.4


RUN

RUN
HISS
```

ACTUAL COUNT

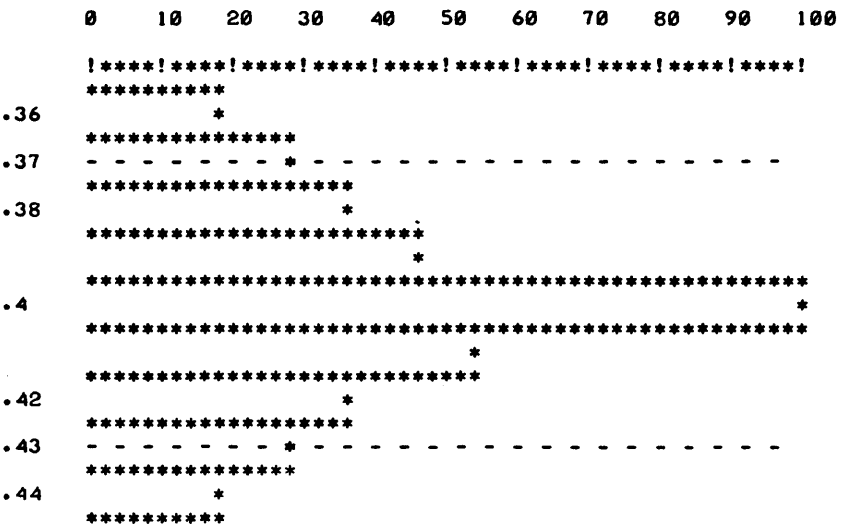
NUMBER OF DATA POINTS= 42 INTERVAL SIZE= .005
MEAN= 45.4001 MEDIAN= 45.4
STANDARD DEVIATION= 2.23738E-02

```
          0         2         4         6         8         10        12
          !*********!*********!*********!*********!*********!*********!
 45.35    ******
 45.355   *
 45.36    ***********
 45.365   *
 45.37    ****************
 45.375   *
 45.38    ********************
 45.385   *
 45.39    *************************
 45.395   *
 45.4     ************************************************************
 45.405   ******
 45.41    *************************
 45.415   *
 45.42    ********************
 45.425   *
 45.43    ****************
 45.435   *
 45.44    ***********
 45.445   *
 45.45    ******
```

```
11 DATA 1,"DEMONSTRATION HISTOGRAM"
12 DATA 1,45.37,45.43
13 DATA 5,45.36,.01,45.44
14 DATA 2
15 DATA 3
16 DATA 3
17 DATA 2

RUN
HISS
```

DEMONSTRATION HISTOGRAM
PERCENT OF MAXIMUM

NUMBER OF DATA POINTS= 42 INTERVAL SIZE= .01
MEAN= 45.4001 MEDIAN= 45.4
STANDARD DEVIATION= 2.23738E-02
LOWER LIMIT= 45.37 UPPER LIMIT= 45.43
THERE IS 1 DATA POINT BELOW 45.36
THERE IS 1 DATA POINT ABOVE 45.44



DONE

CONTRIBUTED PROGRAM **BASIC**

HPMLIT
36218

TITLE: LIST/DUMP HP ASSEMBLER FILES

DESCRIPTION:

These programs used in conjunction with TIDEX, 36204, allow a Time-Share user to build and list and edit a file of assembler code for the HP 2100, and then dump the file to paper tape. The paper tape version is ready for immediate input to any HP assembler. The listed version includes relative addressing.

The package consists of two programs, HPMLIT and HPMLUT.

INSTRUCTIONS:

1. Using TIDEX, 36204, the user builds a file of assembly code.
2. GET-HPMLUT
   100 FILES F1,F2,....
   110 LET X = # of files
3. RUN (This gives an "assembler-like" listing of code with relative addresses
4. Using TIDEX correct any errors
5. GET-HPMLIT
6. 100 FILES F1,F2,....
7. 110 LET X = # of files
8. RUN (dump to paper tape)

SPECIAL CONSIDERATIONS:

These programs are particularly valuable to the HP 2000 user with a CRT or other high speed list output device because it allows for easy on-line corrections before punching a paper tape copy.

ACKNOWLEDGEMENTS: Harry Pyle
Computer Terminal Corporation

```
RUN
GET-HPMLUT
100 FILES SOURCE                        * LISTING
110     X=1
RUN
HPMLUT

00000
00001  ASMB,A,B,L
17123          ORG 17123B
17123          JSB PATC,I
16543          ORG 16543B
16543  PATC    DEF LPDR
16544  LPDR    NOP
16545          STA SAVE,I
16546          LIA 1
16547  SSA,RSS/
16550          SSA,RSS
16551          JMP LPDR,I
16552  LP.1    LDA 102B,I
16553          STA LPDR
16554  LPD     LIA 13B
16555          SZA
16556          JMP *-2
16557  LP.3    JSB GETCH,I
16560          JMP LP.4
16561  LPA     OTA 13B
16562  LPB     STC 13B,C
16563  LPC     SFS 13B
16564          JMP *-1
16565  EN
16566          JMP LP.3
16567  LP.4    CLC 13B
16570          JMP TP.84,I
               END
16571  FOL/

DONE
GET-HPMLIT                              * PUNCHING

100 FILES SOURCE
110   X=1
RUN
HPMLIT



ASMB,A,B,L
       ORG 17123B
       JSB PATC,I
       ORG 16543B
PATC   DEF LPDR
LPDR   NOP
       STA SAVE,I
       LIA 1
SSA,RSS/
       SSA,RSS
       JMP LPDR,I
LP.1   LDA 102B,I
       STA LPDR
LPD    LIA 13B
       SZA
       JMP *-2
LP.3   JSB GETCH,I
       JMP LP.4
LPA    OTA 13B
LPB    STC 13B,C
LPC    SFS 13B
       JMP *-1
EN
       JMP LP.3
LP.4   CLC 13B
       JMP TP.84,I
       END
FOL/


DONE
```

TITLE:

AUTOMATIC PLOTTING PROGRAM

HPPLOT
36805

DESCRIPTION:

HPPLOT enables the time-share and remote computer user, nonprogrammers as well as programmers, to obtain accurate, finished plots easily and quickly on any HP 7200 Series Plotter at his terminal.

The time-share user accustomed to receiving only tabulated output and/or printer plots may now obtain finished line or point graphs and bar or step charts, on his choice of paper (gridded or blank). HPPLOT is a file plotting program. HPPLOT is designed to plot any data from files that may have been created by the user, whether by means of his own program, a library program or from an existing file data base. The data file may contain both alpha and numeric data. The time-share user simply LOADS and RUNS HPPLOT to graph his data.

HPPLOT enables the user to obtain:

1. Point graphs, line graphs, bar charts or step charts.
2. Multiple plots, i.e., a family of "curves" either on the same scale or on different scales.
3. Plots on blank paper or on gridded paper matching his application.
4. Updated plots, i.e., easy "add-ons" to previous plots.
5. Logarithmic or linear plots.
6. Labelled plots with any size letter printed on the horizontal or vertical.

HPPLOT is a program structured to provide a simple yet versatile means to generate graphical presentations of data. HPPLOT may access any data from structured data files. With a few simple commands HPPLOT can generate plots with point, line, bar or step representation of the data points. Scaling may be automatic or specified. Axes are drawn automatically using either log or linear scaling. Axes drawing may be inhibited to plot on pregridded paper. Multiple plots on one axis or different axes may be drawn using multiple colors if desired.

A simple command such as B;LIN/LIN;N/N;A/A would be all the information necessary to produce a Bar Chart with automatic scaling on linear scales drawn and located on the left and bottom of the page.

The program is so structured that it is self-documenting and instructing. Inputs may be entered one at a time or all at once by experienced users. The program allows the axes to be labelled and the graphs to be titled. All aspects of the plot can be controlled by the user so that he may have exactly the type of presentation he desires or the program will automatically plot the data with only one simple command.

This program bridges the gap between the computer's capability for mass data generation and the mind's desire for compact graphical presentation of data. No longer is the time-share user required to switch to manual plotting when it comes to graphing the computer's data. The boring time-consuming job of placing points on a piece of paper is now as easy and convenient as any other routine job the computer performs. The price of the HP 7200 Series Plotters and the ease and versatility of HPPLOT bring graphical capability within the grasp of every time-share user.

ACKNOWLEDGEMENTS:

INSTRUCTIONS:

HPPLOT is **completely** conversational and is specially structured for ease of use and versatility. The HPPLOT basic commands are listed below:

TABLE 1. HPPLOT BASIC COMMANDS

Entry No.	Single Entries	Meaning
1	L	Plot line graphs
	P	Plot point graphs
	B	Draw Bar Charts
	S	Draw Step Charts
2	"Paper number"	Plot according to type of grid and type of scale of the paper specified.
	LIN/LIN	Both X and Y axes linear.
	LIN/LOG	X axis linear and Y axis logarithmic.
	LOG/LIN	X axis logarithmic and Y axis linear.
	LOG/LOG	Both X and Y axes logarithmic.
3	N/N	A new set of X and Y axes (scales) are to be drawn.
	S/S	No new axes are to be drawn. Previous axis set to be used.
	A/A	Another set of both X and Y axes is to be drawn.
	A/S	Another X axis is to be drawn, but same Y axis (scale is to used).
	S/A	Same X axis to be used: Another Y axis to be drawn.
4	A/A	All data values are to be plotted.
	"number" TO "number"/A	Only the X data values in the range "number" TO "number" are to be plotted (e.g., -20 TO 5000 inclusive are to be plotted). No condition is placed on the Y values.
	A/"number"TO"number"	Only Y values in the range "number" TO "number" are to be plotted.
	"number"TO"number"/"number" TO "number"	Only those values that meet both X and Y consitions are to be plotted.

Further instructions are included in HP No. 72050-90002 Field Test Manual.

RUN

```
COMMAND:? DIV=1


COMMAND:? NOSCALE


COMMAND:? RUN
     PLTT
ENTER YOUR INSTRUCTIONS ALL AT ONCE, OR
ENTER P FOR POINTS, L FOR LINES, B FOR BARS OR S FOR STEPS.
? P
ENTER PAPER NUMBER OR THE TYPE OF X AND Y SCALES
? LIN/LIN
ENTER N,S OR A FOR EACH AXIS IN THE FORM:X/Y
ENTER N IF NEW AXIS IS TO BE DRAWN, OR
      S IF SAME AXIS IS TO BE USED, OR
      A IF ANOTHER AXIS IS TO BE DRAWN.
? N/N
ENTER DATA RANGE OF INTEREST, OR
     ENTER A/A IF ALL OF THE DATA ARE TO BE PLOTTED
? 0TO15/0TO100
     PLTL
5000 5000 PLTT
PLTL
0000 9999↑
0000 7500
0000 5000
0000 2500
0000 0000
2000 0000
4000 0000
6000 0000
8000 0000
9999 0000
5000 5000 PLTT
     PLTL
   0 9999↑
 125 9999
5000 5000 PLTT
     PLTL
9999    0↑
9999  125
5000 5000 PLTT
5000 5000 PLTT
     PLTP
 666 5799
1333 6899
1999 7799
2666 8899
3333 9899
3999 8699
4666 7499
5332 6499
5999 7799
6666 8899
7332 7499
7999 6499
5000 5000 PLTT
 12 POINTS PLOTTED  0 POINTS NOT PLOTTED
 0 POINTS OFF SCALE 0 POINTS OUTSIDE RANGE
```

```
COMMAND:? SIZE=3


COMMAND:? TITLE=POINT PLOT
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 5
ENTER Y COORDINATE WHERE FIRST CHARACTER OF THE LABEL IS TO APPEAR
? 50
     PLTL
3332 4999↑
3332 5269
3512 5269
3512 5134
3332 5134
3872 5179↑
3962 5269
3782 5269
3782 4999
3962 4999
3962 5269
4278 4999↑
4368 4999
4323 4999
4323 5269
4278 5269
4368 5269
4683 4999↑
4683 5269
4863 4999
4863 5269
5223 4999↑
5223 5269
5133 5269
5313 5269
6033 4999↑
6033 5269
6213 5269
6213 5134
6033 5134
6483 5269↑
6483 4999
6663 4999
7023 5179↑
7113 5269
6933 5269
6933 4999
7113 4999
7113 5269
7473 4999↑
7473 5269
7383 5269
7563 5269
5000 5000 PLTT
5000 5000 PLTT


COMMAND:? SIZE=2
```

```
COMMAND:? TITLE=P LIN/LIN N/N
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/35
     PLTL
1999 3499↑
1999 3679
2119 3679
2119 3589
1999 3589
2599 3679↑
2599 3499
2719 3499
2929 3499↑
2989 3499
2959 3499
2959 3679
2929 3679
2989 3679
3199 3499↑
3199 3679
3319 3499
3319 3679
3499 3499↑
3619 3679
3799 3679↑
3799 3499
3919 3499
4129 3499↑
4189 3499
4159 3499
4159 3679
4129 3679
4189 3679
4399 3499↑
4399 3679
4519 3499
4519 3679
4999 3499↑
4999 3679
5119 3499
5119 3679
5299 3499↑
5419 3679
5599 3499↑
5599 3679
5719 3499
5719 3679
5000 5000 PLTT
5000 5000 PLTT
```

```
COMMAND:? TITLE=0TO15/0TO100
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/25
     PLTL
1999 2499↑
1999 2679
2119 2679
2119 2499
1999 2499
2359 2499↑
2359 2679
2299 2679
2419 2679
2659 2619↑
2719 2679
2599 2679
2599 2499
2719 2499
2719 2679
2899 2619↑
2959 2679
2959 2499
2899 2499
3019 2499
3319 2679↑
3199 2679
3199 2589
3259 2589
3319 2499
3199 2499
3499 2499↑
3619 2679
3799 2499↑
3799 2679
3919 2679
3919 2499
3799 2499
4159 2499↑
4159 2679
4099 2679
4219 2679
4459 2619↑
4519 2679
4399 2679
4399 2499
4519 2499
4519 2679
4699 2619↑
4759 2679
4759 2499
4699 2499
4819 2499
4999 2499↑
4999 2679
5119 2679
5119 2499
4999 2499
5299 2499↑
5299 2679
5419 2679
5419 2499
5299 2499
5000 5000 PLTT
5000 5000 PLTT
```

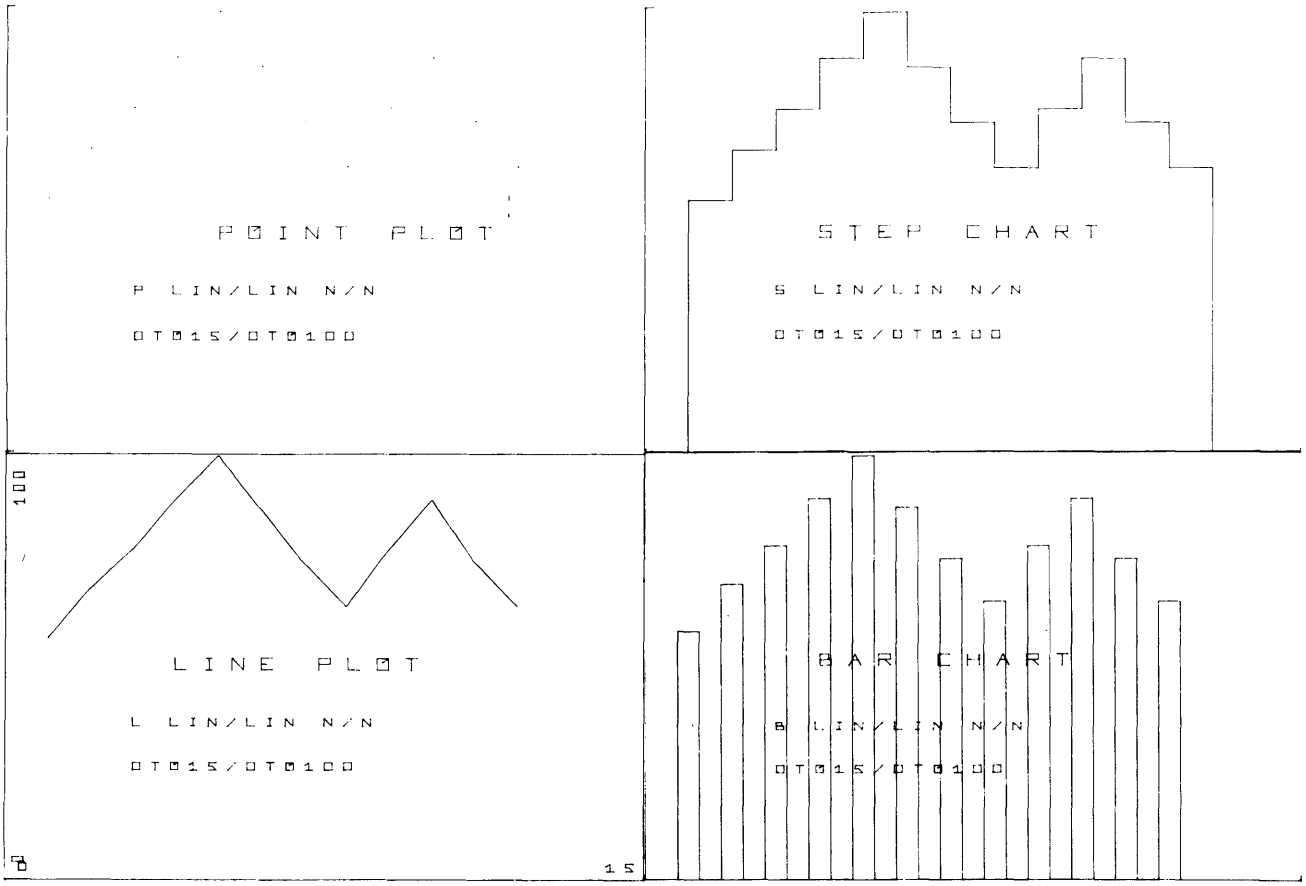
```
COMMAND:? SCALE


COMMAND:? RUN
     PLTT
ENTER YOUR INSTRUCTIONS ALL AT ONCE, OR
ENTER P FOR POINTS, L FOR LINES, B FOR BARS OR S FOR STEPS.
? L;LIN/LIN;N/N;A-0TO15/00-TO100
     PLTL
5000 5000 PLTT
PLTL
0000 9999↑
0000 7500
0000 5000
0000 2500
0000 0000
2000 0000
4000 0000
6000 0000
8000 0000
9999 0000
5000 5000 PLTT
     PLTL
   0 9999↑
 125 9999
5000 5000 PLTT
     PLTL
9999    0↑
9999  125
5000 5000 PLTT
     PLTL
 299  399↑
 119  399
 119  519
 299  519
 299  399
5000 5000 PLTT
     PLTL
 179 8799↑
 119 8859
 299 8859
 299 8799
 299 8919
 299 9099↑
 119 9099
 119 9219
 299 9219
 299 9099
 299 9399↑
 119 9399
 119 9519
 299 9519
 299 9399
5000 5000 PLTT
     PLTL
 224  199↑
 224  379
 344  379
 344  199
 224  199
5000 5000 PLTT
     PLTL
9399  319↑
9459  379
9459  199
9399  199
9519  199
9819  379↑
9699  379
9699  289
9759  289
9819  199
9699  199
5000 5000 PLTT
5000 5000 PLTT
```

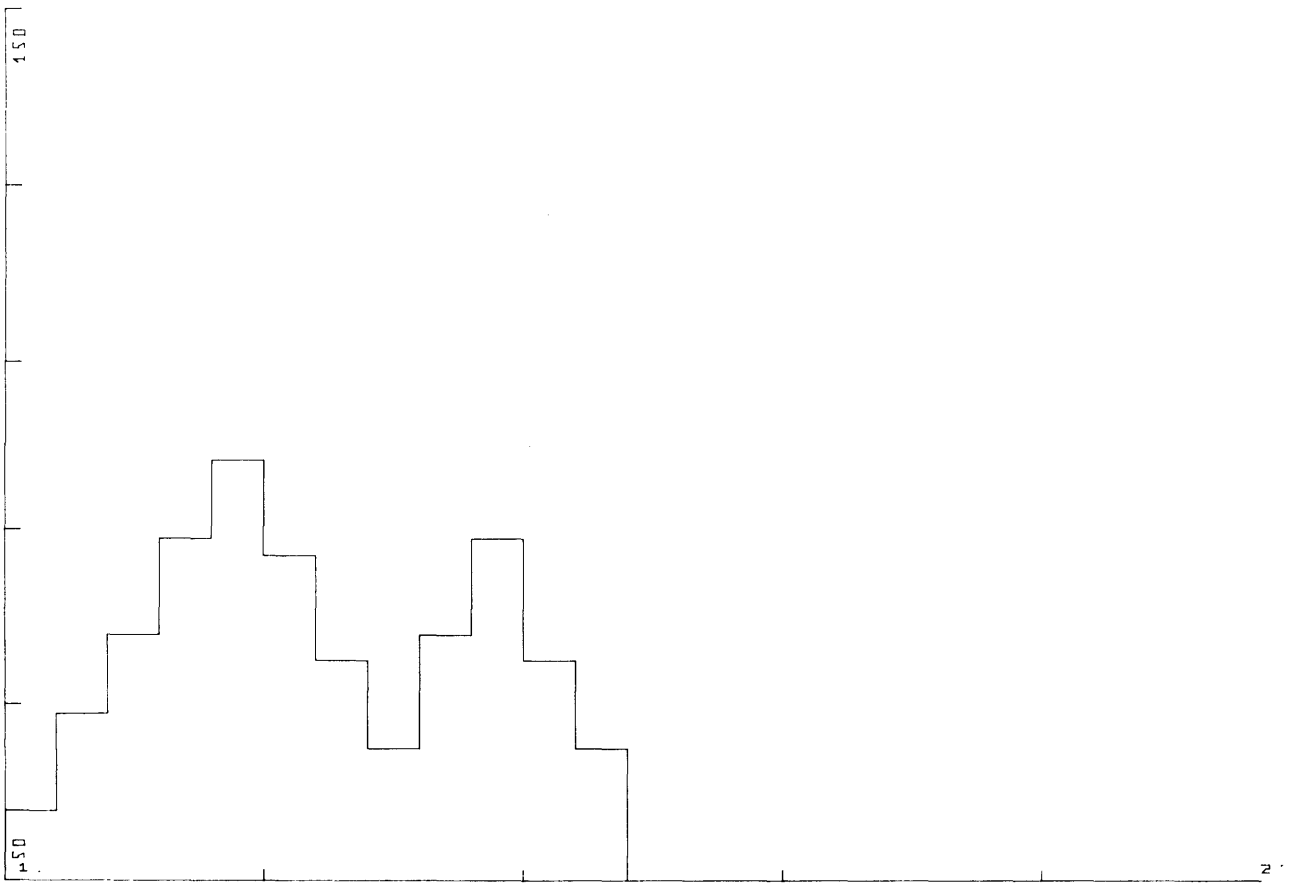
```
     PLTL
 666 5799
1333 6899
1999 7799
2666 8899
3333 9899
3999 8699
4666 7499
5332 6499
5999 7799
6666 8899
6999 8199
7332 7499
7999 6499
5000 5000 PLTT
 12 POINTS PLOTTED  0 POINTS NOT PLOTTED
 0 POINTS OFF SCALE 0 POINTS OUTSIDE RANGE


COMMAND:? SIZE=3


COMMAND:? TITLE=LINE PLOT
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 4/50
     PLTL
5000 5000 PLTT


COMMAND:? SIZ=2


COMMAND:? TITLE=L LIN/LIN N/N
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/35
     PLTL
5000 5000 PLTT


COMMAND:? TITLE=0TO15/0TO100
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/25
     PLTL
5000 5000 PLTT


COMMAND:? NOS


COMMAND:? RUN
     PLTT
ENTER YOUR INSTRUCTIONS ALL AT ONCE, OR
ENTER P FOR POINTS, L FOR LINES, B FOR BARS OR S FOR STEPS.
? S;LIN/LIN;N/N;0TO15/0TO100
     PLTL
PLTL
     PLTL
     PLTL
5000 5000 PLTT
     PLTL
 27 POINTS PLOTTED  0 POINTS NOT PLOTTED
 0 POINTS OFF SCALE 0 POINTS OUTSIDE RANGE


COMMAND:? SIZ=3


COMMAND:? TITLE=STEP CHART
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 4/50
     PLTL
5000 5000 PLTT
```

```
COMMAND:? SIZE=2


COMMAND:? TITLE=S LIN/LIN N/N
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/35
     PLTL
5000 5000 PLTT


COMMAND:? TITLE=0TO15/0TO100
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/25
     PLTL
5000 5000 PLTT


COMMAND:? RUN
     PLTT
ENTER YOUR INSTRUCTIONS ALL AT ONCE, OR
ENTER P FOR POINTS, L FOR LINES, B FOR BARS OR S FOR STEPS.
? B;LIN/LIN;N/N; -0TO15/0TO100
     PLTL
PLTL
     PLTL
     PLTL
5000 5000 PLTT
     PLTL
 61 POINTS PLOTTED  0 POINTS NOT PLOTTED
 0 POINTS OFF SCALE 0 POINTS OUTSIDE RANGE


COMMAND:? TITLE=B LIN/LIN N/N
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/35
     PLTL
1999 3589↑
2119 3589
2119 3499
1999 3499
1999 3679
2089 3679
2089 3589
2599 3679↑
2599 3499
2719 3499
2929 3499↑
2989 3499
2959 3499
2959 3679
2929 3679
2989 3679
3199 3499↑
5000 5000 PLTT


COMMAND:? TITLE=0TO15/0TO100
ENTER X COORDINATE WHERE FIRST CHARACTER OF LABEL IS TO APPEAR
? 3/25
     PLTL
5000 5000 PLTT
```

```
COMMAND:? RUN
     PLTT
ENTER YOUR INSTRUCTIONS ALL AT ONCE, OR
ENTER P FOR POINTS, L FOR LINES, B FOR BARS OR S FOR STEPS.
? B;LIN/LIN;N/N;0TO15/0TO100
     PLTL
5000 5000 PLTT
PLTL
0000 9999↑
0000 7500
0000 5000
0000 2500
0000 0000
2000 0000
4000 0000
6000 0000
8000 0000
9999 0000
5000 5000 PLTT
     PLTL
   0 9999↑
 125 9999
5000 5000 PLTT
     PLTL
9999    0↑
9999  125
5000 5000 PLTT
5000 5000 PLTT
     PLTL
 666 5799
 499 5799
 499 2899
 499    0
 499 2899
 499 5799
 833 5799
 833 2899
 833    0
1166    0
1166 2299
1166 4599
1166 6899
1333 6899
1499 6899
1499 4599
1499 2299
1499    0
1833    0
1833 2599
1833 5199
1833 7799
1999 7799
2166 7799
2166 5199
2166 2599
2166    0
2499    0
2499 2966
2499 5932
2499 8899
2666 8899
2833 8899
2833 5932
2833 2966
2833    0
3166    0
3166 2474
3166 4949
3166 7424
3166 9899
3333 9899
3499 9899
3499 7424
3499 4949
3499 2474
3499    0
3832    0
3832 2899
```

```
3832 5799
3832 8699
3999 8699
4166 8699
4166 5799
4166 2899
4166    0
4499    0
4499 2499
4499 4999
4499 7499
4666 7499
4832 7499
4832 4999
4832 2499
4832    0
5166    0
5166 2166
5166 4332
5166 6499
5332 6499
5499 6499
5499 4332
5499 2166
5499    0
5832    0
5832 2599
5832 5199
5832 7799
5999 7799
6166 7799
6166 5199
6166 2599
6166    0
6499    0
6499 2966
6499 5932
6499 8899
6666 8899
6832 8899
6832 5932
6832 2966
6832    0
7165    0
7165 2499
7165 4999
7165 7499
7332 7499
7499 7499
7499 4999
7499 2499
7499    0
7832    0
7832 2166
7832 4332
7832 6499
7999 6499
8165 6499
8165 4332
8165 2166
8165    0
5000 5000 PLTT
 61 POINTS PLOTTED  0 POINTS NOT PLOTTED
 0 POINTS OFF SCALE 0 POINTS OUTSIDE RANGE
```


POINT PLOT
P LIN/LIN N/N
DT015/DT0100
STEP CHART
S LIN/LIN N/N
DT015/DT0100
100
LINE PLOT
L LIN/LIN N/N
DT015/DT0100
0
15
BAR CHART
B LIN/LIN N/N
DT015/DT0100


150
150
2

CONTRIBUTED PROGRAM **BASIC**

TITLE:

INDEXING PROGRAM

INDEXR
36770

DESCRIPTION:

INDEXR is designed to facilitate the creation of indexes for publications. The entries are entered by page number. As many entries per page as desired may be entered sequentially or in any order. The program produces an alphabetical list of the entries with the page numbers listed in order for each entry.

INSTRUCTIONS:

Instructions are provided in the program. The information is stored on 1, 2, or 3 files of at least 2 records each. An additional scratch file is also required. Thus the program needs at least two files. A files statement is to be entered in line 1 before executing the program. The scratch file is to be the last one. The reason that additional files may be required is to store more information than is allowed in one file. Most applications will require no more than one file of 48 records.

No attempt was made to make the program extremely fast. The entry merely stores the information on the files in a serial fashion. The alphabetic sort makes two passes through the files for each different entry. In most cases only one or two listings will be needed so speed is not a problem.

The only editing allowed is to delete an entire entry be citing it on page "**".

There is no restriction on the form of the page numbers except for "**". For most used forms the program will put them in increasing order.

The output is paginated with hyphen along the left margin exactly 11" apart.

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

Lawrence E. Turner, Jr.
Pacific Union College

RUN

```
OPE-F1,4
OPE-F2,4
GET-INDEXR
1  FILES F1,F2
RUN
INDEXR


               INDEXR

TOTAL NUMBER OF FILES ?2


INSTRUCTIONS ?YES


THIS PROGRAM WAS CREATED TO FACILITATE THE CREATION OF INDEXES FOR
PUBLICATIONS.  ONE ENTERS THE PAGE NUMBER FOLLOWED BY AS MANY ENTRIES
AS DESIRED.  THE PROGRAM IS CONTROLLED BY THE FOLLOWING COMMANDS:

P:     NEW PAGE
O:     OUTPUT
I:     INITIALIZE
S:     STOP

ALL COMMANDS ARE INDICATED WITH A 'CTRL A' AS A FIRST CHARACTER
FOLLOWED BY THE SINGLE LETTER OF THE COMMAND.

ENTRIES MAY BE DELETED BY ENTERING THEM ON PAGE '**'.



?P
PLEASE ENTER PAGE NUMBER ?1.3
?PILOT
?BASIC
?FORTRAN
?SNOBOL
?P
PLEASE ENTER PAGE NUMBER ?A-7
?PILOT
?TERMINAL
?TTY
?CRT
?PORT
?P
PLEASE ENTER PAGE NUMBER ?4
?PILOT
?FORTRAN
?SNAP
?APL
?COBOL
?P
PLEASE ENTER PAGE NUMBER ?1.10
?PILOT
?PYLON
?PL/I
?CAI
?P
PLEASE ENTER PAGE NUMBER ?1.0
?PILOT
?FORTRAN
?BASIC
?IDF
?P
PLEASE ENTER PAGE NUMBER ?1
?PILOT
?ALGOL
?INTERCOM
?MAD
?O
```

INDEX 1

CONTRIBUTED PROGRAM **BASIC**

IRV
36232

TITLE:

FILE SORT ROUTINE

DESCRIPTION:

IRV consists of 2 programs, IRV and IRV 2. These two programs provide a general sort routine for BASIC files. Files are sorted by records, on a specified field within each record. Empty records are automatically skipped. The sort may be on numbers or strings (but not both), in ascending or decending order. Output may be directed to the terminal or another file. Output may be (a) sorted item, (b) record no., (c) sorted item and record number, (d) full record contents, or (e) formatted. No scratch file is used for the sort routine. There is no limit to the number of records which may be sorted.

INSTRUCTIONS:

GET and RUN the program, IRV. The program will ask all necessary questions, and then CHAIN to IRV2, which performs the actual sort. For those users who wish to use a different program with IRV2, it must have a COMMON statement of the following form:

```
1  COM  F$(7), R1, R2, E, T, S, O, P, G$(7), N7, A(10), B(10)
```

where the variables have the following meanings

```
      F$  : Name of file to be sorted.
      R1  : First record to be sorted.
      R2  : Last record to be sorted.
       E  : Which item in each record is to be sorted.
       T  :  = 1 : each item is a number
             = 2 : each item is a string
       S  :  = 1 : sort is ascending
             = 2 : sort is descending
       O  : Output desired
             = 1 : sorted item
             = 2 : record number
             = 3 : sorted item and record number
             = 4 : full record contents
             = 5 : formatted output
       P  :  = 1 : output to terminal
             = 2 : output to a file
      G$  : name of output file (if any).
      N7  : no. of formatted strings to be output for
            each sorted record, (if any).
       A  : array containing the item numbers of the
            formatted strings to be output.
       B  : array containing the column numbers the formatted
            strings must begin on.
```

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

RUN

```
RUN
IRV

ARE YOU AN EXPERT WITH THIS PROGRAM?NO

WHAT FILE DO YOU WANT TO SORT?NAME

DO YOU WANT TO SORT ON ALL RECORDS?NO
  ON WHAT RECORD DO YOU WANT TO START THE SORT?1
  ON WHAT RECORD DO YOU WANT TO END THE SORT?10

ON WHICH ELEMENT OF EACH RECORD DO YOU WANT TO SORT?1

IS THIS ELEMENT A NUMBER OR A STRING IN EACH RECORD OF THE FILE?STRING

DO YOU WANT THE SORT IN ASCENDING OR DESCENDING ORDER?ASCENDING

SPECIFY YOUR OUTPUT:

  1 = SORTED ITEM
  2 = RECORD NUMBER
  3 = SORTED ITEM AND RECORD NUMBER
  4 = FULL RECORD CONTENTS
  5 = FORMATTED OUTPUT

OUTPUT DESIRED (1-5)?5
TYPE THE LIST OF ITEMS YOU WANT PRINTED,
EACH FOLLOWED BY THE COLUMN YOU WANT IT TO BEGIN IN.
TO END THE LIST, TYPE '0,0'.

ITEM NUMBER, COLUMN?2,1
ITEM NUMBER, COLUMN?3,8
ITEM NUMBER, COLUMN?4,14
ITEM NUMBER, COLUMN?1,20
ITEM NUMBER, COLUMN?0,0


FILENAME: NAME          RECORDS 1     THROUGH 10


DCSA    NSR    NOV    ADAMS, JOEL
SE      ESR    NOV    BARTLETT,MARY
SA      ESR    NOV    BILLINGS,IRV
SA      SSR    NOV    BRENNER,BRUCE
DCSA    NSR    NOV    EDWARDS,DIANE
SE      MSR    NOV    EDWARDS,SUSAN
SA      SSR    NOV    MILLER,SCOTT
SE      ESR    NOV    PEARCE,REBECCA
DCSA    MSR    NOV    WALLACE,CLAYTON
SE      NSR    NOV    WOODS,JOEL


SORT COMPLETED.

DONE


RUN
IRV

ARE YOU AN EXPERT WITH THIS PROGRAM?NO

WHAT FILE DO YOU WANT TO SORT?NAME

DO YOU WANT TO SORT ON ALL RECORDS?NO
  ON WHAT RECORD DO YOU WANT TO START THE SORT?3
  ON WHAT RECORD DO YOU WANT TO END THE SORT?15

ON WHICH ELEMENT OF EACH RECORD DO YOU WANT TO SORT?1

IS THIS ELEMENT A NUMBER OR A STRING IN EACH RECORD OF THE FILE?STRING

DO YOU WANT THE SORT IN ASCENDING OR DESCENDING ORDER?ASCENDING
```

```
SPECIFY YOUR OUTPUT:

  1 = SORTED ITEM
  2 = RECORD NUMBER
  3 = SORTED ITEM AND RECORD NUMBER
  4 = FULL RECORD CONTENTS
  5 = FORMATTED OUTPUT

OUTPUT DESIRED (1-5)?3

DO YOU WANT THE SORT PRINTED ON YOUR TERMINAL, OR ON A FILE?TERMINAL


FILENAME: NAME          RECORDS 3     THROUGH 15


RECORD    ITEM


 3     BILLINGS,IRV
 4     BRENNER,BRUCE
 5     EDWARDS,DIANE
 6     EDWARDS,SUSAN
 11    LOCKWOOD,JANE
 13    MCCOOL,JIM
 12    MCNEIL,PAT
 7     MILLER,SCOTT
 14    MONTGOMERY,ALAN
 8     PEARCE,REBECCA
 15    REED,HARRY
 9     WALLACE,CLAYTON
 10    WOODS,JOEL


SORT COMPLETED.

DONE
```

CONTRIBUTED PROGRAM **BASIC**

JULIAN
36197

TITLE: JULIAN CALENDAR FOR THE CURRENT YEAR

DESCRIPTION:

This program generates a Julian calendar for the current year. This calendar is useful to owners of Time-Share Systems since following a sleep the system requires the Julian date. It will generate Julian calendars for each year from 1971 to 1995. The only input required by the program is the calendar year desired.

INSTRUCTIONS:

When "RUN" the program will execute and ask for all four digits of current year.

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

Ray Agrusti
Omega Data Processing

RUN

```
RUN
JULIAN

PROGRAM TO GENERATE JULIAN CALENDAR FOR CURRENT YEAR


ENTER ALL FOUR DIGITS OF CURRENT YEAR?1972


JULIAN CALENDAR                                                    1972

JANUARY

SAT- 1    - 1    SUN- 2    - 2    MON- 3    - 3    TUE- 4    - 4
WED- 5    - 5    THU- 6    - 6    FRI- 7    - 7    SAT- 8    - 8
SUN- 9    - 9    MON- 10   - 10   TUE- 11   - 11   WED- 12   - 12
THU- 13   - 13   FRI- 14   - 14   SAT- 15   - 15   SUN- 16   - 16
MON- 17   - 17   TUE- 18   - 18   WED- 19   - 19   THU- 20   - 20
FRI- 21   - 21   SAT- 22   - 22   SUN- 23   - 23   MON- 24   - 24
TUE- 25   - 25   WED- 26   - 26   THU- 27   - 27   FRI- 28   - 28
SAT- 29   - 29   SUN- 30   - 30   MON- 31   - 31


FEBRUARY

TUE- 1    - 32   WED- 2    - 33   THU- 3    - 34   FRI- 4    - 35
SAT- 5    - 36   SUN- 6    - 37   MON- 7    - 38   TUE- 8    - 39
WED- 9    - 40   THU- 10   - 41   FRI- 11   - 42   SAT- 12   - 43
SUN- 13   - 44   MON- 14   - 45   TUE- 15   - 46   WED- 16   - 47
THU- 17   - 48   FRI- 18   - 49   SAT- 19   - 50   SUN- 20   - 51
MON- 21   - 52   TUE- 22   - 53   WED- 23   - 54   THU- 24   - 55
FRI- 25   - 56   SAT- 26   - 57   SUN- 27   - 58   MON- 28   - 59
TUE- 29   - 60


MARCH

WED- 1    - 61   THU- 2    - 62   FRI- 3    - 63   SAT- 4    - 64
SUN- 5    - 65   MON- 6    - 66   TUE- 7    - 67   WED- 8    - 68
THU- 9    - 69   FRI- 10   - 70   SAT- 11   - 71   SUN- 12   - 72
MON- 13   - 73   TUE- 14   - 74   WED- 15   - 75   THU- 16   - 76
FRI- 17   - 77   SAT- 18   - 78   SUN- 19   - 79   MON- 20   - 80
TUE- 21   - 81   WED- 22   - 82   THU- 23   - 83   FRI- 24   - 84
SAT- 25   - 85   SUN- 26   - 86   MON- 27   - 87   TUE- 28   - 88
WED- 29   - 89   THU- 30   - 90   FRI- 31   - 91


APRIL

SAT- 1    - 92   SUN- 2    - 93   MON- 3    - 94   TUE- 4    - 95
WED- 5    - 96   THU- 6    - 97   FRI- 7    - 98   SAT- 8    - 99
SUN- 9    - 100  MON- 10   - 101  TUE- 11   - 102  WED- 12   - 103
THU- 13   - 104  FRI- 14   - 105  SAT- 15   - 106  SUN- 16   - 107
MON- 17   - 108  TUE- 18   - 109  WED- 19   - 110  THU- 20   - 111
FRI- 21   - 112  SAT- 22   - 113  SUN- 23   - 114  MON- 24   - 115
TUE- 25   - 116  WED- 26   - 117  THU- 27   - 118  FRI- 28   - 119
SAT- 29   - 120  SUN- 30   - 121


JULIAN CALENDAR                                                    1972

MAY

MON- 1    - 122  TUE- 2    - 123  WED- 3    - 124  THU- 4    - 125
FRI- 5    - 126  SAT- 6    - 127  SUN- 7    - 128  MON- 8    - 129
TUE- 9    - 130  WED- 10   - 131  THU- 11   - 132  FRI- 12   - 133
SAT- 13   - 134  SUN- 14   - 135  MON- 15   - 136  TUE- 16   - 137
WED- 17   - 138  THU- 18   - 139  FRI- 19   - 140  SAT- 20   - 141
```

SUN- 21	- 142	MON- 22	- 143	TUE- 23	- 144	WED- 24	- 145
THU- 25	- 146	FRI- 26	- 147	SAT- 27	- 148	SUN- 28	- 149
MON- 29	- 150	TUE- 30	- 151	WED- 31	- 152		

JUNE

THU- 1	- 153	FRI- 2	- 154	SAT- 3	- 155	SUN- 4	- 156
MON- 5	- 157	TUE- 6	- 158	WED- 7	- 159	THU- 8	- 160
FRI- 9	- 161	SAT- 10	- 162	SUN- 11	- 163	MON- 12	- 164
TUE- 13	- 165	WED- 14	- 166	THU- 15	- 167	FRI- 16	- 168
SAT- 17	- 169	SUN- 18	- 170	MON- 19	- 171	TUE- 20	- 172
WED- 21	- 173	THU- 22	- 174	FRI- 23	- 175	SAT- 24	- 176
SUN- 25	- 177	MON- 26	- 178	TUE- 27	- 179	WED- 28	- 180
THU- 29	- 181	FRI- 30	- 182				

JULY

SAT- 1	- 183	SUN- 2	- 184	MON- 3	- 185	TUE- 4	- 186
WED- 5	- 187	THU- 6	- 188	FRI- 7	- 189	SAT- 8	- 190
SUN- 9	- 191	MON- 10	- 192	TUE- 11	- 193	WED- 12	- 194
THU- 13	- 195	FRI- 14	- 196	SAT- 15	- 197	SUN- 16	- 198
MON- 17	- 199	TUE- 18	- 200	WED- 19	- 201	THU- 20	- 202
FRI- 21	- 203	SAT- 22	- 204	SUN- 23	- 205	MON- 24	- 206
TUE- 25	- 207	WED- 26	- 208	THU- 27	- 209	FRI- 28	- 210
SAT- 29	- 211	SUN- 30	- 212	MON- 31	- 213		

AUGUST

TUE- 1	- 214	WED- 2	- 215	THU- 3	- 216	FRI- 4	- 217
SAT- 5	- 218	SUN- 6	- 219	MON- 7	- 220	TUE- 8	- 221
WED- 9	- 222	THU- 10	- 223	FRI- 11	- 224	SAT- 12	- 225
SUN- 13	- 226	MON- 14	- 227	TUE- 15	- 228	WED- 16	- 229
THU- 17	- 230	FRI- 18	- 231	SAT- 19	- 232	SUN- 20	- 233
MON- 21	- 234	TUE- 22	- 235	WED- 23	- 236	THU- 24	- 237
FRI- 25	- 238	SAT- 26	- 239	SUN- 27	- 240	MON- 28	- 241
TUE- 29	- 242	WED- 30	- 243	THU- 31	- 244		

JULIAN CALENDAR 1972

SEPTEMBER

FRI- 1	- 245	SAT- 2	- 246	SUN- 3	- 247	MON- 4	- 248
TUE- 5	- 249	WED- 6	- 250	THU- 7	- 251	FRI- 8	- 252
SAT- 9	- 253	SUN- 10	- 254	MON- 11	- 255	TUE- 12	- 256
WED- 13	- 257	THU- 14	- 258	FRI- 15	- 259	SAT- 16	- 260
SUN- 17	- 261	MON- 18	- 262	TUE- 19	- 263	WED- 20	- 264
THU- 21	- 265	FRI- 22	- 266	SAT- 23	- 267	SUN- 24	- 268
MON- 25	- 269	TUE- 26	- 270	WED- 27	- 271	THU- 28	- 272
FRI- 29	- 273	SAT- 30	- 274				

OCTOBER

SUN- 1	- 275	MON- 2	- 276	TUE- 3	- 277	WED- 4	- 278
THU- 5	- 279	FRI- 6	- 280	SAT- 7	- 281	SUN- 8	- 282
MON- 9	- 283	TUE- 10	- 284	WED- 11	- 285	THU- 12	- 286
FRI- 13	- 287	SAT- 14	- 288	SUN- 15	- 289	MON- 16	- 290
TUE- 17	- 291	WED- 18	- 292	THU- 19	- 293	FRI- 20	- 294
SAT- 21	- 295	SUN- 22	- 296	MON- 23	- 297	TUE- 24	- 298
WED- 25	- 299	THU- 26	- 300	FRI- 27	- 301	SAT- 28	- 302
SUN- 29	- 303	MON- 30	- 304	TUE- 31	- 305		

NOVEMBER

WED- 1	- 306	THU- 2	- 307	FRI- 3	- 308	SAT- 4	- 309
SUN- 5	- 310	MON- 6	- 311	TUE- 7	- 312	WED- 8	- 313
THU- 9	- 314	FRI- 10	- 315	SAT- 11	- 316	SUN- 12	- 317
MON- 13	- 318	TUE- 14	- 319	WED- 15	- 320	THU- 16	- 321
FRI- 17	- 322	SAT- 18	- 323	SUN- 19	- 324	MON- 20	- 325
TUE- 21	- 326	WED- 22	- 327	THU- 23	- 328	FRI- 24	- 329
SAT- 25	- 330	SUN- 26	- 331	MON- 27	- 332	TUE- 28	- 333
WED- 29	- 334	THU- 30	- 335				

DECEMBER

FRI- 1	- 336	SAT- 2	- 337	SUN- 3	- 338	MON- 4	- 339
TUE- 5	- 340	WED- 6	- 341	THU- 7	- 342	FRI- 8	- 343
SAT- 9	- 344	SUN- 10	- 345	MON- 11	- 346	TUE- 12	- 347
WED- 13	- 348	THU- 14	- 349	FRI- 15	- 350	SAT- 16	- 351
SUN- 17	- 352	MON- 18	- 353	TUE- 19	- 354	WED- 20	- 355
THU- 21	- 356	FRI- 22	- 357	SAT- 23	- 358	SUN- 24	- 359
MON- 25	- 360	TUE- 26	- 361	WED- 27	- 362	THU- 28	- 363
FRI- 29	- 364	SAT- 30	- 365	SUN- 31	- 366		

DONE

CONTRIBUTED PROGRAM **BASIC**

TITLE:

FILE LOAD/DUMP

LODUMP
36644

DESCRIPTION:

This program creates maximally compacted paper tape copies of serial files and provides for the reloading of such tapes. Files can also be loaded via compatible manually prepared tapes or keyboard input.

INSTRUCTIONS:

1. Dumping files to tape: Input file name and "dump" option as requested by the program and turn on tape punch when indicated.

2. Loading files from tape: Input file name and "load" option. Program asks if you wish to add data to the end of the data currently on the file. If you respond with "NO" then the new data is written over the old.

3. Manual preparation of tapes: (a) Items to be loaded on a file are separated by one or more blank spaces. (b) Any items that corresponds to a standard numeric format, i.e., integer, real, or exponential, is interpreted as a number. (c) Strings containing blank spaces are delineated by enclosing them with the back slash character, "\". (d) Each line must end with "X-OFF" - "RETURN" - "LINEFEED". (e) To terminate input the last line should contain only "CONTROL C"- "X-OFF" - "RETURN".

4. Keyboard file loading: To load files on line via the teletype keyboard follow the instructions for manual tape preparation, omitting the "X-OFF" at the end of each line.

SPECIAL CONSIDERATIONS:

1. To input via paper tape the terminal must have a tape reader that responds to "X-OFF" and "X-ON".

2. The back slash is a special character and cannot be present in any strings to be loaded or dumped.

3. Maximum string size allowed is 71 characters.

ACKNOWLEDGEMENTS:

William W. Moss
University of Virginia School of Medicine

```
RUN

RUN
LODUMP


FILE NAME ?AFILE

LOAD OR DUMP ?LOAD

DO YOU WANT TO ADD TO THE EXISTING FILE ?NO

START TAPE READER OR MANUAL INPUT


THIS IS A TEST    OF    LODUMP
THESE    WORDS ARE   TREATED  AS SEPARATE STRINGS
AS ARE   THESE/WHILE←←←←←←\WHILE THESE ARE NOT   \
NUMBERS AND STRINGS CAN 1 1.234 678.945 37878383.38983 BE -12E21
6.57E17 MIXED 345 LIKE 3.21E-8 THIS 4.5675E+30
\TERMINAL BACKSLASH IS NOT REQUIRED IF NO MORE DATA IS ON THE LINE
\STRINGS THAT LOOK LIKE NUMERIC FORMATS CAN BE ENTERED BY
\INCLUDING THEM WITHIN BACKSLASHES WITH EITHER A BLANK SPACE\
\OR A NONPRINTED CHARACTER SUCH AS CONTROL G (BELL)\ FOR EXAMPLE
123 IS A NUMBER WHILE \ 123\ IS A STRING AS IS 123
(NOTE THAT THE LAST STRING CONTAINED A NONPRINTED CHARACTER)
\DATA INPUT IS ENDED BY ENTERING A CONTROL C BY ITSELF\

DONE


RUN
LODUMP


FILE NAME ?AFILE

LOAD OR DUMP ?DUMP

TURN ON PUNCH



THIS IS A TEST OF LODUMP THESE WORDS ARE TREATED AS SEPARATE STRINGS AS
ARE THESE\WHILE THESE ARE NOT  NUMBERS AND STRINGS CAN 1 1.234 678.945
3.78784E+07 BE -1.2E+22 6.57E+17 MIXED 345 LIKE 3.21E-08 THIS
4.5675E+30
\TERMINAL BACKSLASH IS NOT REQUIRED IF NO MORE DATA IS ON THE LINE\
\STRINGS THAT LOOK LIKE NUMERIC FORMATS CAN BE ENTERED BY\
\INCLUDING THEM WITHIN BACKSLASHES WITH EITHER A BLANK SPACE\
\OR A NONPRINTED CHARACTER SUCH AS CONTROL G (BELL)\FOR EXAMPLE 123 IS
A NUMBER WHILE \ 123\IS A STRING AS IS 123 (NOTE THAT THE LAST STRING
CONTAINED A NONPRINTED CHARACTER)
\DATA INPUT IS ENDED BY ENTERING A CONTROL C BY ITSELF\


DONE



RUN
LODUMP


FILE NAME ?AFILE

LOAD OR DUMP ?LOAD

DO YOU WANT TO ADD TO THE EXISTING FILE ?YES

START TAPE READER OR MANUAL INPUT


THIS IS A TEST OF LODUMP THESE WORDS ARE TREATED AS SEPARATE STRINGS AS
ARE THESE\WHILE THESE ARE NOT  NUMBERS AND STRINGS CAN 1 1.234 678.945
3.78784E+07 BE -1.2E+22 6.57E+17 MIXED 345 LIKE 3.21E-08 THIS
4.5675E+30
```

```
\TERMINAL BACKSLASH IS NOT REQUIRED IF NO MORE DATA IS ON THE LINE\
\STRINGS THAT LOOK LIKE NUMERIC FORMATS CAN BE ENTERED BY\
\INCLUDING THEM WITHIN BACKSLASHES WITH EITHER A BLANK SPACE\
\OR A NONPRINTED CHARACTER SUCH AS CONTROL G (BELL)\FOR EXAMPLE 123 IS
A NUMBER WHILE \ 123\IS A STRING AS IS 123 (NOTE THAT THE LAST STRING
CONTAINED A NONPRINTED CHARACTER)
\DATA INPUT IS ENDED BY ENTERING A CONTROL C BY ITSELF\

DONE


RUN
LODUMP


FILE NAME ?AFILE

LOAD OR DUMP ?DUMP

TURN ON PUNCH



THIS IS A TEST OF LODUMP THESE WORDS ARE TREATED AS SEPARATE STRINGS AS
ARE THESE\WHILE THESE ARE NOT  NUMBERS AND STRINGS CAN 1 1.234 678.945
3.78784E+07 BE -1.2E+22 6.57E+17 MIXED 345 LIKE 3.21E-08 THIS
4.5675E+30
\TERMINAL BACKSLASH IS NOT REQUIRED IF NO MORE DATA IS ON THE LINE\
\STRINGS THAT LOOK LIKE NUMERIC FORMATS CAN BE ENTERED BY\
\INCLUDING THEM WITHIN BACKSLASHES WITH EITHER A BLANK SPACE\
\OR A NONPRINTED CHARACTER SUCH AS CONTROL G (BELL)\FOR EXAMPLE 123 IS
A NUMBER WHILE \ 123\IS A STRING AS IS 123 (NOTE THAT THE LAST STRING
CONTAINED A NONPRINTED CHARACTER)
\DATA INPUT IS ENDED BY ENTERING A CONTROL C BY ITSELF\
THIS IS A TEST OF LODUMP THESE WORDS ARE TREATED
A ARE THESE\WHILE THESE ARE NOT NUMBERS AND STRINGS CAN 1 1.234 678.945
3.78784E+07 BE -1.2E+22 6.57E+17 MIXED 345 LIKE 3.21E-08 THIS
4.5675E+30
\TERMINAL BACKSLASH IS NOT REQUIRED IF NO MORE DATA IS ON THE LINE\
\STRINGS THAT LOOK LIKE NUMERIC FORMATS CAN BE ENTERED BY\
\INCLUDING THEM WITHIN BACKSLASHES WITH EITHER A BLANK SPACE\
\OR A NONPRINTED CHARACTER SUCH AS CONTROL G (BELL)\FOR EXAMPLE 123 IS
A NUMBER WHILE \ 123\IS A STRING AS IS 123 (NOTE THAT THE LAST STRING
CONTAINED A NONPRINTED CHARACTER)
\DATA INPUT IS ENDED BY ENTERING A CONTROL C BY ITSELF\


DONE
```

CONTRIBUTED PROGRAM **BASIC**

MACRO
36003B

TITLE: A TEXT AND FILE PROCESSING SYSTEM

DESCRIPTION:

This program allows editing any file on a 2000 series system with no loss of file integrity. This editor also has macro capabilities allowing for creation of user defined commands.

INSTRUCTIONS:

See Attached.

SPECIAL CONSIDERATIONS:

The file being edited may have only one EOF mark, and each record within the file must have at least one data element on it other than an EOR. The control-Z, when used as a delimiter in a macro or a multiple command line, causes the cursor on the 2600A keyboard display to move up one line. This causes problems when typing or listing macros as some data on the screen is destroyed. If you have a 2600A, you should change line 9171 so that the 'IF BRANCH' will occur on some other character of your choice. This character will then become your multiple command and macro delimiter. This statement causes a branch on control-Z.

For operating this program on systems where disc record size <>64, make the following changes. Substitute the record size of your system for RS.

```
9029  If L < RS+1 THEN 9008
9084  If L < RS+1 THEN 9087
9091  If L < RS+1 THEN 9094
9375  If L > RS   THEN 9570
```

ACKNOWLEDGEMENTS: Joel Bartlett
HP, Data Systems Division

MACRO

A TEXT AND FILE PROCESSING SYSTEM

A few notation conventions:

n = a number

[and] enclose optional quantities

{ and } enclose quantities from which one must chose one.

INTRODUCTION

There has been a need for a comprehensive file management program. This program provides these facilities and also provides text handling capabilities for the HP 2000 series time sharing systems. It is fully compatible with the 2000A system.

MACRO allows the user to create data files of any form and content. Once created, either by this program or by another program, a file may be edited. In doing so, file elements may be referenced by their sequential position from the head of the file, or by their record number and position in that record. This allows both sequential and random access files to be edited with equal facility. List, insertion, deletion, and replacement commands; find and substitute commands; and character editing commands are some of the MACRO commands which are useful in data file handling. A file may also be restructured using commands to block it into fixed length records, or to pack it into a sequential file.

A closely related area is text handling. Here MACRO may be used to create text files, and then manipulate them. Test may be inserted, replaced, deleted, or edited on a character-by-character basis. Also, facilities exist for string location and string substitution. Finally, there exist commands that produce justified text, margins, and clean output copy.

BASIC CONCEPTS

A few concepts must first be introduced in order to allow understanding of MACRO's principles, and thereby gain ease of use. The basic data element is either a number, or a string within the file. Let the following file be an example:

```
RECORD 1
 12.7
 33.6
 AAAA
RECORD 2
 16.9
 BCB
 17.3
END OF FILE
```

This file contains 6 data elements, at 3 per record. Each data element in turn has both a sequence (position) number associated with it, and a record number and sequence (in the record) number associated with it. Either of these may be used to reference a given element. Here is an example of access by sequence number only: the string "BCB" is number five (5) under the sequential reference method. Using

the second method of reference by both record and sequence number, the same string would be referenced as record 2, element 2; (2,2). The following example shows the relation between the two numbering systems using the same sample file.

RECORD #1	SEQUENCE NUMBER	RECORD AND SEQUENCE ON THE RECORD
12.7	1	1,1
33.6	2	1,2
AAAA		
RECORD #2		
16.9	4	2,1
BCB	5	2,2
17.3	6	2,3
END OF FILE		

One of the main concepts of MACRO is that of the "current datum", referred to as the C.D. in this manual. The C.D. is that element in the file to which a pointer is currently set. Most commands either act on the C.D., or the C.D. and data elements which follow it. At the end of each operation, the pointer is reset, usually to a different data element. It is thus important that the user be aware of the effect of his commands on the C.D., especially when using the macro commands. This information is found in the explanation of each command, and is also summarized in a table in the appendix.

TO USE MACRO

To use MACRO, create a one-line BASIC program with a sequence number under 9000. For example:

```
600 FILES file1, file2
```

file1 is the name of the file that you wish to create or edit. Your new or corrected file will be under this name on a legal exit.

file2 is a temporary editor scratch file. This file must be at least as large as file1.

Both files must have had storage allocated for them using the system command OPEN before MACRO is RUN.

When this is done, type the following commands:

```
APP-$MACRO
RUN
```

After "RUN" is typed, MACRO will ask "OLD OR NEW FILE?". The user then responds with "OLD" or "NEW". If he typed "NEW", C.D. will be set to 1> last element in the file and data may now be added to the file at the rate of one data element per line. If the user desires a number to be entered as a string, he should precede it with a control-A(A^C). To advance to the next record of the file, type a vertical arrow (↑). Finally, to end the file, type a line consisting solely of a control-Z(Z^C). At this point, an END OF FILE mark will be written onto the new file, and the C.D. will be set to the first data element on the file, with the file ready for editing.

Below is an example showing how one could have created the earlier shown sample file using MACRO:

```
OPEN-EX,2
OPEN-T*,2
10 FILES EX,T*
```

```
APP-$MACRO
RUN
OLD OR NEW FILE?NEW
-?12.7
-?33.6
-?AAAA
-?↑
-?16.9
-?BCB
-?17.3
-?Z^C
2  =LAST WRITTEN RECORD
>?
```

If "OLD" was answered in response to the "OLD OR NEW FILE" query, then the C.D. would have been set to the first data element of the file, with the file ready for editing.

MACRO MODES

The following sections describe the system's commands. These commands may be entered, unless otherwise specified, when MACRO types a ">?". In response to an I or R command or when creating a new file, a "-?" will be typed. This means that the system is waiting for new data to be entered. This can be seen in the example on the preceeding page. After the A command is given, MACRO types a "*?" whenever it expects another A sub-group command. A "$?" will be typed only when a macro definition is expected. These special characters will hopefully aid the user in remembering the commands and proper responses. They are summarized in the appendix, in a table which again describes these modes and then shows which commands may be entered after which symbol.

BASIC COMMANDS

This section introduces the user to the basic commands of the system. Each command is an abbreviation for a word. It will aid the user in remembering the command if he will associate the command with the word from which it is derived. To aid this association, these "keywords" are underlined and capitalized in each command explanation, and are noted in the appendix.

One of the most important commands is the E command, which effects an orderly EXIT from MACRO. This command insures that all the edits made will be executed and that the resultant file will be left in file1. The command's format is:

```
>?E
DONE
```

It is important not to use the break key or a control-C (C^C) to exit from MACRO as some edits may be lost and the final file may not be in file1. If for some reason either of these two keys is used for exit, a recovery may be effected using the crash procedures explained later.

COMMANDS WHICH AFFECT THE CURRENT DATUM

Since most commands address relative to the current datum (C.D.), it is important to be able to use the C.D. positioning commands.

The CR command resets the pointer to the new C.D. which is specified either by its sequence number, or by both a RECORD number and a sequence number on that record. This command allows absolute addressing of file elements. Its format is:

```
>?CR n,n
```

For example, the following command would set the C.D. to the fourth element of the second record:

```
>?CR 2,4
```

and this command would set the C.D. to the eighteenth member of the file:

```
>?CR 18
```

Another command sets the new C.D. by ADDRESSING RELATIVE to the present C.D. Its format is:

```
>?C{+}n
   {-}
```

Thus if the next element of the file is to become the new C.D., the command C+1 will reset the C.D. pointer to it. Likewise if one wished to set the C.D. to a data element that occurs previous to the C.D. then one would use the C- command.

There also exists a command which sets the C.D. at either end of the file. It is the C$ command which sets the C.D. at either the BEGINNING of the file or at the END of the file. Its format is:

```
>?C${B}
    {E}
```

The B option sets the C.D. to the first data element of the file, and the E option sets the C.D. to the last element of the file.

The C command causes the CURRENT C.D. to be printed on the terminal. Its format is:

```
>?C
```

Record and sequence numbers may or may not be printed, depending upon the status of the PRINT FLAG (see the PF command). The C.D., if numeric, will be printed with a preceeding "N".

Below is an example showing the use of the commands that have been explained at this point.

```
GET-MACRO
10 FILES E1,E2
RUN
MACRO

OLD OR NEW FILE?NEW
-?THIS IS THE FIRST LINE OF RECORD 1
-?12
-?13
-?14
-?↑
-?THIS IS THE FIRST LINE OF RECORD 2
-?22
```

```
-?23
-?24
-?
 2     =LAST WRITTEN RECORD
>?C
 1     THIS IS THE FIRST LINE OF RECORD 1     R= 1
>?C$E
>?C
 8     N  24       R= 2
>?CR  2,1
>?C
 5     THIS IS THE FIRST LINE OF RECORD 2     R= 2
>?CR  5
>?C
 5     THIS IS THE FIRST LINE OF RECORD 2     R= 2
>?C$B
>?C
 1     THIS IS THE FIRST LINE OF RECORD 1     R= 1
>?C+2
>?C
 3     N  13       R= 1
>?C-1
>?C
 2     N  12       R= 1
>?E
DONE
```

OUTPUT COMMANDS

A second group of commands affect output and output format. None of these commands affect the setting of the pointer to the C.D.

The PF command sets the PRINT FLAG which determines whether the sequence and record numbers of the C.D. are to be printed along with the data when the C command is given. The command format is:

>?PF[{R / S}]

PF R causes both record and sequence numbers to be printed along with the C.D. PF S causes only the sequence number to be printed with the C.D., and PF causes only the C.D. to be printed. Here is an example with the C.D. being a line whose sequence number from the head of the file is 16, and is found on record #2.

```
  >?PF R
  >?C
16    LINE 16, FOUND ON RECORD TWO  R=2
  >?PF S
  >?C
16    LINE 16, FOUND ON RECORD TWO
  >?PF
  >?C
LINE 16, FOUND ON RECORD TWO
```

The MG command, whose format is:

>?MG n

sets a lefthand MARGIN for all output. The parameter n in the command is the number of blanks that will be inserted between the lefthand print position of the terminal, and the first character to be printed. On entrance to the editor, the margin is set at zero.

The L command causes the printing of a LIST of the file from the current datum through a specified range at the rate of one data element per line. Its format is:

$$>?L \left\{ \begin{matrix} \$E \\ n \end{matrix} \right\} \quad \left[\left\{ \begin{matrix} R \\ S \end{matrix} \right\} \right]$$

Note here, that this command has two parameter fields, with the second field being optional. The first parameter specified the number (if n) of data elements to be listed. This number includes the C.D. It may also be a "$E" in which case data (including the C.D.) will be listed until an end of file is reached. The second parameter specifies whether sequence numbers (S) or sequence and record numbers (R) should be listed along with the data. This use of "S" and "R" is identical to their use in the PF command. The L command is also like the C command in that numeric data will be printed with a leading "N".

A second form of the L command is used to LIST a MACRO (see the advanced features section for more on macros). Its form is:

>?LM n (1 > =n > =3)

This command will cause that numbered macro to be listed on the terminal.

The following example shows the use of this group of commands.

```
10 FILES E1,E2
APP-MACRO
RUN

OLD OR NEW FILE?OLD
>?C
 1    THIS IS THE FIRST LINE OF RECORD 1      R= 1
>?PF S
>?C
 1    THIS IS THE FIRST LINE OF RECORD 1
>?PF
>?C
THIS IS THE FIRST LINE OF RECORD 1
>?L $E
THIS IS THE FIRST LINE OF RECORD 1
N  12
N  13
N  14
THIS IS THE FIRST LINE OF RECORD 2
N  22
N  23
N  24
>?L 2 R
 1    THIS IS THE FIRST LINE OF RECORD 1      R= 1
 2    N  12       R= 1
>?L 3 S
```

```
 1    THIS IS THE FIRST LINE OF RECORD 1
 2    N  12
 3    N  13
>?MG 10
>?L$E
          THIS IS THE FIRST LINE OF RECORD 1
          N  12
          N  13
          N  14
          THIS IS THE FIRST LINE OF RECORD 2
          N  22
          N  23
          N  24
>?E
DONE
```

DATA EDITING COMMANDS

The third and final group of the basic commands handle the actual editing functions with the use of the C.D. as their reference point.

The I command INSERTS new data elements after the C.D. Its format is:

```
>?I
-?
```

The new data is added exactly as the data is added on a new file. When the Z^C is entered, the insertion is completed, and the new C.D. is set at the old C.D.+k+1, where k is the number of new data elements that have been added. This is shown in the example which follows this section.

The second command of this group is R. This command REPLACES the C.D. with one or more data elements. Its format is:

```
>?R
-?
```

The replacement operation is identical to the new data addition for I. On completion of the replacement (a Z^C was entered), the new C.D. is set at the old C.D.+k, where k is the number of new data elements entered.

In both of the above commands, there is no limitation on the data types of the new elements.

The D command DELETES the C.D. Its format is:

```
>?D
```

Following the execution of this command, the file pointer is left unchanged so that the new C.D. is the next element in the file.

Below is an example showing the use of these three commands.

```
RUN

OLD OR NEW FILE?NEW
-?THIS IS THE FIRST LINE
-?AND THIS IS THE SECOND LINE.
```

```
-?AND THIS IS THE THIRD LINE.
-?FINALLY THIS IS THE LAST LINE.
-?
 1    =LAST WRITTEN RECORD
>?I
-?******************
-?
>?R
-?$$$$$$$$$$$$$$$$$$$$$4
-?
>?D
>?C$B
>?L $E
THIS IS THE FIRST LINE
******************
$$$$$$$$$$$$$$$$$$$$4
FINALLY THIS IS THE LAST LINE.
>?I
-?$$$$$$$$$$$$$$$$$$$
-?####################
-?####################
-?
>?C$B
>?L$E
THIS IS THE FIRST LINE
$$$$$$$$$$$$$$$$$$
##################
##################
**********
$$$$$$$$$$$$$$$$$$4
FINALLY THIS IS THE LAST LINE.
>?E
DONE
```

The A command (for ALTER) is the last member of this group. It allows individual character editing on the C.D. (if it is a string type datum) by putting the user into alter mode. The command format is:

```
>?A
```

An attempt at altering numeric data will result in the printing of a diagnostic message which leaves the C.D. unchanged. When this command is entered, the system will print out the C.D. and then print a "*?" signifying that it is ready to accept one of the following sub-commands.

E	This command EXITS the user from alter mode. The C.D. is then set to the next data element of the file.
C	This command causes printing of the C.D. in its current state of editing.
P character n	This command POSITIONS the first occurrence of the given character at the numbered position in the line (left-most character =1). This command is mainly used for such things as centering titles.
S character	This command SKIPS a character pointer to the first occurrence of the given character in a line, and then prints out a vertical arrow to show the position of the pointer. To set the pointer at later occurrences of a character in the line,

repeat the command. To reset this pointer before the beginning of the line, give the command with a character not in the line. This command must be used before character insertion, replacement, or deletion can occur.

I character string

This command inserts the given character string after the character which the pointer indicates.

R character string

This command replaces the character denoted by the pointer with the given character string.

After the use of either the I command or the R command, the S command must be used to reset the character pointer before another I,R, or D command will be accepted.

D This deletes the character pointed to by the line pointer. The pointer is set to the following character in the line after the execution of this command.

Below is an example involving these alter sub-group commands.

```
OLD OR NEW FILE?NEW
-?THIS IS THE ONLY LINE OF THE FILE, EXCEPT FOR THE SECOND.
-?THIS IS THE SECOND LINE OF THE FILE.
-?
 1    =LAST WRITTEN RECORD
>?A
THIS IS THE ONLY LINE OF THE FILE, EXCEPT FOR THE SECOND.
*?S,
*?I*****
*?C
THIS IS THE ONLY LINE OF THE FILE,***** EXCEPT FOR THE SECOND.
*?S.
*?R.
*?C
THIS IS THE ONLY LINE OF THE FILE,***** EXCEPT FOR THE SECOND.
*?S*
*?D
*?D
*?D
*?D
*?D
*?C

THIS IS THE ONLY LINE OF THE FILE, EXCEPT FOR THE SECOND.
*?SN
*?S
*?D
*?PT 10
*?C
          THIS IS THE ONLY LINE OF THE FILE, EXCEPT FOR THE SECOND.
*?E
>?C
 2     THIS IS THE SECOND LINE OF THE FILE.     R= 1
>?E
DONE
```

FILE ORGANIZATION AND INTEGRITY

The MACRO system is designed so that the integrity of a record-oriented file is preserved. However, the system cannot prevent the user from overflowing a record during the editing process. When this occurs, the record overflow will be contained on another record directly following the overflowed record. The original contents of that record are moved to the next record and so forth through the rest of the file. Thus, the overflow record contains overflow and nothing else. The following example shows the use of the I command to force this overflow.

```
 1     **************************************************     R= 1
 2     ************************     R= 1
 3     **************************************************     R= 2
 4     ************************     R= 2
>?CR 2
>?I
-?$$$$$$$$$$$$$$$$$$$$$$$4
-?$$$$$$$$$$$$$$$$$$$$$$$$
-?
>?C$B
>?L $E R
 1    **************************************************     R= 1
 2    ************************    R= 1
 3    $$$$$$$$$$$$$$$$$$$$$$$4    R= 1
 4    $$$$$$$$$$$$$$$$$$$$$$$$    R= 2  ←  OVERFLOW RECORD
 5    **************************************************     R= 3
 6    ************************    R= 3
>?E
```

There is only one limitation in the type of file that this system will edit. It is the fact that the file can have only one end of record mark. This is a requirement, as MACRO cannot differentiate between a physical end of file and a software end of file.

Because of its ability to handle record-oriented files, optimum data packing does not always occur in a sequential file. The following command can rectify this situation.

```
>?PS
```

This command causes optimum packing of data elements on a sequential basis. A related command controls the blocking of data into fixed length records. Its form is:

```
>?PR n
```

This command causes the file to be packed into records with n data elements per record. Diagnostic messages will be printed on the terminal in the case of file or record overflow. When either of these commands is finished execution, the C.D. is set to the first element of the file.

TEXT EDITING FEATURES

The J command causes a file consisting solely of strings to be left JUSTIFIED word by word into lines less than or equal to a specified length. This command will not break up strings of blanks. Its format is:

```
>?J n          where n specifies the line length
```

Two control words, "$LINE" and "$PAGE" exist to produce blank lines and paging when listing text. $LINE causes 1 blank line to be spaced, and $PAGE causes 6 blank lines to be spaced. These control words are

only converted into linefeeds under two conditions, first when the C command is given when the PF command has preceded it, and when the L command is used without the S or R options. If a line starts with one of these control words, it will be expanded into linefeeds under these conditions. The J command recognizes these control words only if they are alone in a line (i.e. individual string) and will then not pack them into lines of fixed length. Thus, when wishing to use these control words for formatting a text file, one must use an entire line for them.

ADVANCED EDITING FEATURES

FIND COMMANDS

These commands search the C.D. and the following data in the file for the first occurrence of a given substring or number. When the substring or number is found, the C.D. is set to that element of the file. If on the other hand, an end of file is encountered before the substring or number is found, then the search is stopped and the C.D. is set to the first data element of the file.

The NF (NUMBER FIND) command searches for a particular number in the file. Its form is:

>NF n where n is the number that the user wishes to find.

The SF (STRING FIND) command is identical to the NF command except that it searches for a particular substring in the file. A match will occur if this string equals one in the file, or if this substring is a substring of some string in the file. Its format is:

>SF character string

Any spaces following the SF are considered to be part of the string for which the user wishes to find a match.

SUBSTITUTE COMMANDS

These commands are identical to the find commands except that once the datum is found, a substitution is made as directed by the command. The C.D. is then set to the datum that is found after the substitution is made. If no match was found, then the C.D. is set to the first data element of the file, and the search is stopped.

The NSUB (NUMBER SUBSTITUTE) command will substitute one number for another. Its form is:

>?NSUB $n_1 \backslash n_2$

Here, n_1 will be substituted for the first occurrence of n_2, The backslash character (\ ) which can be read as "for" is the character shift-L on the terminal.

The second substitute command (STRING SUBSTITUTE) will substitute one substring for another. Its format is:

>?SSUB $s_1 \backslash s_2$

Here, s_1 is substituted for the first occurrence of s_2. Any spaces occurring after the SSUB will be considered to be either part of s_1, if on the left side of the backslash, or part of s_2 if they occur to the right of the backslash.

MULTIPLE COMMANDS

Several commands may be written on one line in the following manner:

```
>? C$B ZC PF ZC C ZC A
```

Note the use of the Z^C's to separate the individual commands. The only commands allowed in such a multiple command line are those which may be entered after a "?." If an unrecognized command occurs in a string of commands, the error will be noted and control of the editing process will return to the terminal keyboard. The following example shows the use of a line of commands to justify, margin and list a piece of text and then exit from MACRO.

```
>?J 20 MG 25 L$E E
                        THIS IS THE FIRST
                        LINE OF A FILE THAT
                        WILL BE JUSTIFIED,
                        AND THEN LISTED
                        USING THE MULTIPLE
                        COMMAND FORMAT OF
                        THE EDITING SYSTEM,
                        'MACRO'.
DONE
```

The following example shows the use of the find and substitute commands.

```
>?C$B
>?L$E
THIS IS THE FIRST LINE OF A NEW FILE.
N  12
N  24
N  32
AND THIS IS THE SECOND LINE OF IT.
FINALLY THIS IS THE THIRD LINE OF THIS FILE.
>?NF 24
>?C
 3    N  24
>?C$B
>?SFFINALLY
>?C
 6    FINALLY THIS IS THE THIRD LINE OF THIS FILE.
>?C$B
>?SSUB*\
>?C
 1    THIS*IS*THE*FIRST*LINE*OF*A*NEW*FILE.
>?SSUB\ *
>?C
 1    THIS IS THE FIRST LINE OF A NEW FILE.
>?NSUB 200\12
>?C
 2    N  200
>?C$B
>?L$E
```

```
THIS IS THE FIRST LINE OF A NEW FILE.
N 200
N 24
N 32
AND THIS IS THE SECOND LINE OF IT.
FINALLY THIS IS THE THIRD LINE OF THIS FILE.
>?E
DONE
```

MACRO COMMANDS

macro: a user-defined command embodying several system or user defined commands so that they will be executed when the macro is called.

Many times when editing, one would like to repeat a specific set of commands without retyping them all. For this reason, some limited macro capabilities have been implemented in this editor. This following group of commands are related to macros. They do not affect the C.D. in any manner.

The DEF command is used to DEFINE macros. Its format is:

```
>DEF n          n is between 1 and 3 inclusive
```

The user is allowed three macros, numbered 1 to 3 inclusive. When the "$?" is typed by the system, the user enters his macro definition as a string of commands separated by Z^C's, just as in the case of multiple commands. A return is used to terminate the definition. All commands except those belonging to the A sub-group may appear in a macro definition.

The single quote command (') causes the text which follows it to be printed out on the terminal. When this command is inside a macro, no leading spaces may appear before the quote ('). Its format is:

```
>?' any characters excluding a Z^C or inside a macro
 $?...Z^C'  HELLO!!Z^C...
```

The MACRO system has an instruction TRACE to aid in debugging macros. It is enabled and disabled by the .T. command. This command complements a flag such that every odd occurrence of this instruction will actuate the trace, and every even numbered occurrence of the command will clear this trace flag. Upon entry to MACRO, this flag is initialized such that no trace will occur. The command format is:

```
>?.T.
```

The command may also appear inside a macro.

The .M. command (format .M.) allows DECISION MAKING in a MACRO while it is executing. The .M. command causes the system to ask "NEXT MACRO?". To this, the user has two possible responses. He may type in the number of the next macro that he wishes executed, or he may type the letter "Q" which will return the user to keyboard control of the editing process.

The MAC command calls a previously defined MACRO into EXECUTION. Its format is:

```
>MAC n          n is between 1 and 3 inclusive
```

This command may be keyboard entered or be found in the body of a macro. It is important to realize that both this command and the .M. command result in an unconditional transfer of control to a macro.

Macros may also be written in infinite loops with no ill effects. Any error condition or end of file will result in control being returned to the keyboard. For a complete list of conditions which will terminate a macro's execution, see the appendix.

EXAMPLES

This section contains some examples showing the use of most commands in macros for certain file operations.

The first example uses a simple, non-repetitive macro to set a margin, justify a file, and then list it.

```
>?DEF  1
$?MG  1Ø J 5Ø L$E E
>?MAC  1

          A NUMBER OF LANGUAGES ARE BASED ON THE CONCEPT
          KNOWN AS POLISH NOTATION; THIS HAS ADVANTAGES FOR
          MACHINE CODES BUT IS DIFFICULT FOR HUMAN
          DIGESTION.  IT IS SO CALLED BECAUSE IT WAS FIRST
          INTRODUCED BY THE POLISH PHILOSOPHER LUKASIEWICZ
          IN CONNECTION WITH THE FORMULAE OF SYMBOLIC
          LOGIC.  A VARIATION MORE PROPERLY CALLED 'REVERSE
          POLISH' IS MORE POPULAR TODAY IN COMPUTING
          CIRCLES.

DONE
```

The second example shows the use of a macro which calls itself. The purpose of this macro is to list a file from tail to head. Note here, how control returned to the keyboard when the command C-1 tried to execute with the C.D. being the first element of the file. Before control returned to the keyboard, MACRO set the C.D. to the first element of the file, and then printed "HOF" signifying that the C.D. was the element at the HEAD OF the FILE.

```
>?PS
>?DEF 1
$?PF C C-1 MAC 1
>?C$E
>?MAC 1
CIRCLES.
POLISH' IS MORE POPULAR TODAY IN COMPUTING
LOGIC.  A VARIATION MORE PROPERLY CALLED 'REVERSE
IN CONNECTION WITH THE FORMULAE OF SYMBOLIC
INTRODUCED BY THE POLISH PHILOSOPHER LUKASIEWICZA
DIGESTION.  IT IS SO CALLED BECAUSE IT WAS FIRST
MACHINE CODES BUT IS DIFFICULT FOR HUMAN
KNOWN AS POLISH NOTATION; THIS HAS ADVANTAGES FOR
A NUMBER OF LANGUAGES ARE BASED ON THE CONCEPT
HOF
>?
```

The third example involves the use of 3 macros and the .M. command so as to allow string substitution on keyboard approval. To start, the C.D. is the first element of the file, and the first macro is then

called. When the system types "NEXT MACRO?", the user would type a 2 if he wanted the substitution to take place, or a 3 if he did not want the substitution. Also in this example is a macro to do an unqualified substitution over a whole file, and print each string after it makes the substitution.

```
>?DEF 2
$?SSUB*\  C+1 MAC 1
>?DEF 3
$?C+1 MAC 1
>?C$B

>?DEF 1
$?SF  C .M.
>?MAC 1
A NUMBER OF LANGUAGES ARE BASED ON THE CONCEPT
NEXT MACRO?2
KNOWN AS POLISH NOTATION; THIS HAS ADVANTAGES FOR
NEXT MACRO?3
MACHINE CODES BUT IS DIFFICULT FOR HUMAN
NEXT MACRO?2
DIGESTION.  IT IS SO CALLED BECAUSE IT WAS FIRST
NEXT MACRO?3
INTRODUCED BY THE POLISH PHILOSOPHER LUKASIEWICZ
NEXT MACRO?2
IN CONNECTION WITH THE FORMULAE OF SYMBOLIC
NEXT MACRO?3
LOGIC.  A VARIATION MORE PROPERLY CALLED 'REVERSE
NEXT MACRO?2
POLISH' IS MORE POPULAR TODAY IN COMPUTING
NEXT MACRO?3
CIRCLES.
NEXT MACRO?2
HOF
>?L $E
A*NUMBER*OF*LANGUAGES*ARE*BASED*ON*THE*CONCEPT*
KNOWN AS POLISH NOTATION; THIS HAS ADVANTAGES FOR
MACHINE*CODES*BUT*IS*DIFFICULT*FOR*HUMAN*
DIGESTION.  IT IS SO CALLED BECAUSE IT WAS FIRST
INTRODUCED*BY*THE*POLISH*PHILOSOPHER*LUKASIEWICZ*
IN CONNECTION WITH THE FORMULAE OF SYMBOLIC
LOGIC.*A*VARIATION*MORE*PROPERLY*CALLED*'REVERSE*
POLISH' IS MORE POPULAR TODAY IN COMPUTING
CIRCLES.*
>?DEF 1
$?SSUB\ * C MAC 1
>?MAC 1
A NUMBER OF LANGUAGES ARE BASED ON THE CONCEPT
MACHINE CODES BUT IS DIFFICULT FOR HUMAN
INTRODUCED BY THE POLISH PHILOSOPHER LUKASIEWICZ
LOGIC.  A VARIATION MORE PROPERLY CALLED 'REVERSE
CIRCLES.'
```

```
          A NUMBER OF LANGUAGES ARE BASED ON THE CONCEPT
          KNOWN AS POLISH NOTATION; THIS HAS ADVANTAGES FOR
          MACHINE CODES BUT IS DIFFICULT FOR HUMAN
          DIGESTION.  IT IS SO CALLED BECAUSE IT WAS FIRST
          INTRODUCED BY THE POLISH PHILOSOPHER LUKASIEWICA
          IN CONNECTION WITH THE FORMULAE OF SYMBOLIC
          LOGIC.  A VARIATION MORE PROPERLY CALLED 'REVERSE
          POLISH' IS MORE POPULAR TODAY IN COMPUTING
          CIRCLES.
>?E
```

The final example shows the use of the MACRO system as a system for data storage and retrieval. The sample here uses a small group of HP field sales offices. Using this data and macros; listings may be made of these offices by state, region, or country. Note also, the use of .T. for an instruction trace in this example.

```
10 FILES E1, E2
APP-MACRO
RUN

OLD OR NEW FILE?OLD
>?L $E S
 1    $LINE EASTERN WESTERN MIDWESTERN SOUTHERN UNITED STATES
 2    $LINE EASTERN WESTERN MIDWESTERN SOUTHERN UNITED STATES
 3              HP UNITED STATES SALES OFFICES
 4              HP SOUTHERN REGION SALES OFFICES
 5              HP WESTERN REGION SALES OFFICES
 6              HP MIDWESTERN REGION SALES OFFICES
 7              HP EASTERN REGION SALES OFFICES
 8    $LINE SOUTHERN WESTERN EASTERN MIDWESTERN UNITED STATES
 9    ALABAMA, HUNTSVILLE              SOUTHERN, UNITED STATES
 10   ARIZONA, SCOTTSDALE              WESTERN, UNITED STATES
 11   ARIZONA, TUCSON                  WESTERN, UNITED STATES
 12   CALIFORNIA, NORTH HOLLYWOOD      WESTERN, UNITED STATES
 13   CALIFORNIA, PALO ALTO            WESTERN, UNITED STATES
 14   CALIFORNIA, SACRAMENTO           WESTERN, UNITED STATES
 15   CALIFORNIA, SAN DIEGO            WESTERN, UNITED STATES
 16   COLORADO, ENGLEWOOD              WESTERN, UNITED STATES
 17   CONNECTICUT, EAST HARTFORD       EASTERN, UNITED STATES
 18   CONNECTICUT, NORWALK             EASTERN, UNITED STATES
 19   DELAWARE, WILMINGTON             EASTERN, UNITED STATES
 20   ILLINOIS, SKOKIE                 MIDWESTERN, UNITED STATES
 21   $PAGE EASTERN WESTERN MIDWESTERN SOUTHERN UNITED STATES
>?PF
>?DEF 1
$?SFUNITED C C+1 MAC 1

          HP UNITED STATES SALES OFFICES
ALABAMA, HUNTSVILLE              SOUTHERN, UNITED STATES
ARIZONA, SCOTTSDALE              WESTERN, UNITED STATES
```

```
ARIZONA, TUCSON                 WESTERN, UNITED STATES
CALIFORNIA, NORTH HOLLYWOOD     WESTERN, UNITED STATES
CALIFORNIA, PALO ALTO           WESTERN, UNITED STATES
CALIFORNIA, SCRAMENTO           WESTERN, UNITED STATES
CALIFORNIA, SAN DIEGO           WESTERN, UNITED STATES
COLORADO, ENGLEWOOD             WESTERN, UNITED STATES
CONNECTICUT, EAST HARTFORD      EASTERN, UNITED STATES
CONNECTICUT, NORWALK            EASTERN, UNITED STATES
DELAWARE, WILMINGTON            EASTERN, UNITED STATES
ILLINOIS, SKOKIE                MIDWESTERN, UNITED STATES


HOF
>?DEF 1
$?SFMIDW C C+1 MAC 1
>?MAC 1

          HP MIDWESTERN REGION SALES OFFICES
ILLINOIS, SKOKIE                MIDWESTERN, UNITED STATES


HOF
>?DEF 1
$?SF WEST C C+1 MAC 1
>?MAC 1

          HP WESTERN REGION SALES OFFICES

ARIZONA, SCOTTSDALE             WESTERN, UNITED STATES
ARIZONA, TUCSON                 WESTERN, UNITED STATES
CALIFORNIA, NORTH HOLLYWOOD     WESTERN, UNITED STATES
CALIFORNIA, PALO ALTO           WESTERN, UNITED STATES
CALIFORNIA, SACRAMENTO          WESTERN, UNITED STATES
CALIFORNIA, SAN DIEGO           WESTERN, UNITED STATES
COLORADO, ENGLEWOOD             WESTERN, UNITED STATES

HOF
>?DEF 1
$?SF SOUTH C C+1 MAC 1

          HP SOUTHERN REGION SALES OFFICES

ALABAMA, HUNTSVILLE             SOUTHERN, UNITED STATES

HOF
>?DEF 1
$?SF EAST C C+1 MAC 1
>?MAC 1

          HP EASTERN REGION SALES OFFICES

CONNECTICUT, EAST HARTFORD      EASTERN, UNITED STATES
CONNECTICUT, NORWALK            EASTERN, UNITED STATES
DELAWARE, WILMINGTON            EASTERN, UNITED STATES
```

```
HOF
>?DEF 1
$?SFCALIF C C+1 MAC 1
>?MAC 1
CALIFORNIA, NORTH HOLLYWOOD     WESTERN, UNITED STATES
CALIFORNIA, PALO ALTO           WESTERN, UNITED STATES
CALIFORNIA, SACRAMENTO          WESTERN, UNITED
CALIFORNIA, SAN DIEGO           WESTERN, UNITED STATES
>?.T.
>?MAC 1
**-MAC1
**-SFCALIF
**-C
CALIFORNIA, NORTH HOLLYWOOD     WESTERN, UNITED STATES
**-C+1
**-MAC1
**-SFCALIF
**-C
CALIFORNIA, PALO ALTO           WESTERN, UNITED STATES
**-C+1
**-MAC1
**-SFCALIF
**-C
CALIFORNIA, SACRAMENTO          WESTERN, UNITED STATES
**-C+1
**-MAC1
**-SFCALIF
**-C
CALIFORNIA, SAN DIEGO           WESTERN, UNITED STATES
**-C+1
**-MAC1
**-SFCALIF
>?C
```

ERROR MESSAGES

FILE OVERFLOW, TERMINAL ERROR
: MACRO terminates as the edited file is too large to fit in either of the two files.

INPUT REQUIRED PARAMETER
: A command has been entered without a required numeric parameter. Enter the parameter and the editing will continue.

EOF, FILE CLOSED
: The file of new data is full. Input of data for the file is terminated. Editing may now be done on the file.

NULL FILE
: A file does not have any data on it.

ONLY STRINGS MAY BE ALTERED
: The C.D. is numeric and the user has tried to use the A command.

'S', 'C', 'P', OR 'E' COMMAND EXPECTED
: The user is in alter mode and he does not have his character pointer set, hence he is not allowed to use the I, R, or D commands. This diagnostic usually comes about when one tries to make two insertions or replacements in a row.

RESULTANT STRING OVER 72 CHARACTERS

This diagnostic occurs when the resultant string for an I or R in alter mode is over 72 characters long. The new string will not be made, and the system will ask for another A sub-group command.

COMMAND ERROR — Unrecognized command.

STRING OVERFLOW n — A string substitution cannot be made in line n as the resultant string would be over 72 characters in length. The SSUB command continues its search for another match and a possible substitution.

ERROR, NUMERIC INFORMATION ON FILE

This is printed when one attempts to justify a file that contains one or more numeric data elements.

EOF OR EOR ON n_1;n_2 BEFORE COMPLETION

Out of file or record space on file n_1, record n_2 during the execution of a PS, PR, or J command.

ON MACRO SYSTEM CRASH OR ILLEGAL EXIT

In case of a system crash or illegal exit, a user's file is not lost if he follows these restart instructions:

1. Use the system library program FILIST to list both files.
2. If the most recent edits are in the first file, then the user's file is in tact and he may continue using MACRO or any other program.
3. If on the other hand, the most recent edits are in file 2 (MACRO temporary), then the user should use the system library program FLCOPY to copy the temporary file into the user's original file.

Another abort condition occurs when the user has added too much data to his file, and as a result his whole file will not fit in either his original file or his temporary file. In this case, one file will contain all the old file, and the other file will contain part of the edited file. To restart from this exit, do the following:

1. Use the system library program FILIST to list both files. Decide which file to keep, and open a third file of the same size or larger.
2. Use the system library program FLCOPY to copy this file (the one the user is keeping) into the third file.
3. Kill (using the KILL command) the first two files, and then open (using the OPEN command) then to a larger size.
4. Use FLCOPY to copy the third file into the new first file.

CONDITIONS WHICH WILL TERMINATE THE EXECUTION OF A MACRO OR A MULTIPLE COMMAND LINE

1. End of macro or command line.
2. End of file while trying to access the new C.D. after completion of a command. When this occurs, the C.D. will be set to the first element of the file, and the letters "HOF" will be printed on the terminal.

3. End of file while executing a SF,SSUB, NF, or NSUB command. The C.D. will then be set to the first element of the file.
4. On an unrecognized command.
5. On an attempt to reference a non-existent datum (i.e. the datum with the sequence number of zero). The results of this will be identical to those for #2 above.
6. When an end of file or end of record condition occurs in executing a PS, PR, or J command.

COMMAND OVERVIEW

COMMAND	KEYWORD	C.D. AFTER EXECUTION	MODE ALLOWED AFTER
A	ALTER	C.D.+1	>?
C	print C.D.	no effect on C.D.	>? or *?
C^{+}_{-}	CHANGE C.D.	changed as specified	>?
C$	ENDS of file	changed as specified	>?
CR	set C.D. by RECORD	changed as specified	>?
D	DELETE	next datum on file, or next character when in alter mode.	>? or *?
DEF	DEFINE	unchanged	>?
E	EXIT	C.D.+1 when in alter mode. Exits user from program otherwise	>? or *?
I	INSERT	C.D.+n+1 where n is the number of data elements added to the file.	>? or *?
J	JUSTIFY	first data element of the file.	>?
L	LIST	no effect	>?
MAC	MACRO	no effect	>?
MG	MARGIN	no effect	>?
NF	NUMBER FIND	first occurrence of number, otherwise the first element of the file if an end of file is encountered during search.	>?
NSUB	NUMBER	at first substitution	>?
	SUBSTITUTE	or first data element in the file if no substitution could be made	
PF	PRINT FLAG	no effect on C.D.	>?
PR	PACK RECORD	set to first data element on the file	>?
PS	PACK	first data element	>?
	SEQUENTIAL	of file	
R	REPLACE	C.D.+k, where k is the number of new data elements	>?
SF	STRING FIND	see NF	>?

COMMAND	KEYWORD	C.D. AFTER EXECUTION	MODE ALLOWED AFTER
SSUB	STRING SUBSTITUTE	see NSUB	>?
'	QUOTE	no effect on C.D.	>?
.T.	TRACE	no effect on C.D.	>?
.M.	Next MACRO	no effect on C.D.	>?

SPACES: spaces are only critical in the following commands; A sub-group, ', SF, and SSUB. Any other command may have spaces freely intermixed in it. However, this group of commands does not allow leading spaces, and spaces enclosed in the command are critical.

MACROS: All commands that may occur after a ">?" may be in a macro. The maximum length of a macro is 72 characters, including non-printing control characters.

ALTER SUB-GROUP COMMANDS:

C	print CURRENT
D	DELETE a character
E	EXIT from alter mode
I	INSERT a character string
P	POSITION a line
R	REPLACE a character with a character string
S	SKIP the character pointer to a specific character

These commands are entered in response to a "*?". Each time an I or R command is executed, the character pointer is cleared. It must then be reset using the S command. The D command sets the character pointer to the next character in the line when it is finished executing.

SPECIAL CHARACTERS:

←	This character deletes the last typed character. It may be repeated to delete more characters. For example: "ABC←←E←" is equivalent to "A"
escape/altmode	A blackslash (\ ) will be printed on the terminal and the whole line that has been typed will be scratched.
return	This character is used to end each line of command(s) or each line of data.
↑	This character advances the user to the next record of his file when he is adding new text to a file.
Z^{C*}	This character is used in two areas: 1. as a delimiter between commands in a macro or in a multiple command line, 2. and as a terminator for the addition of new data, when creating a new file, or editing an old file.
A^{C*}	This character is used, when adding new data to a file, to allow a number to be entered as a string. The A^{C} must be the first character input, and it will be stripped off before the string is written onto the file.

*control characters do not print on the teletype terminal.

A COMMAND SUBSET

Do not be alarmed by the many commands of MACRO. They are there to aid you, not confuse you. The following is a list of commands that would constitute a basic subset. With these commands, one can perform any editing operation. They are:

CR	to set the C.D.
D	to delete the C.D.
E	to exit from MACRO
I	to insert new data
L	to list part or all of the file
R	to replace the C.D.

It is suggested that the user learn these commands and use them. When the user feels a need for the more powerful commands, he should use them, but not until. It should also be noted that with this command set, the user cannot do serious damage to his file while learning. This claim cannot be made for any of the other commands.

THE PROGRAM

LENGTH: 4359 WORDS
NUMBERING: 9000-9999 BY 1
PROGRAMMER: JOEL BARTLETT AUG 1970

VARIABLES USED: A$,B$,D$,E$,F$,G$,H$,M$,T$,Z$
A1,A2,A3,A4,A5,C4,C9,D,D1,F1, F2
F3, F4, F9, I, I1, I9, J, K, L, L1, M,
M9, P, P9, R, S, T, T6, T7, T8, T9

INTERNAL LAYOUT:

STATEMENT RANGE	DESCRIPTION
9000-9003	String declarations, initialization, and transfer to the program start.
9004-9032	Routine to open file to a given record and element on that record, or to a given sequence number from the head of the file. This routine also transfers data elements to F2 to set up the new file. The variable J is used to keep track of the record number on the two files, and the variable L is used to keep track of the length into the currently being read record. This routine also has a second entry point at 9008 that is used to get the next item on file F1.
9033-9042	This section completes the transfer from F1 to F2 after a command has been executed. It then swaps the file numbers to reflect the new file.
9043-9053	This is the compare routine for the find and substitute commands.
9054-9118	This section puts the new data onto file F2. Record integrity is preserved through the use of variables L and J as set by the routine at 9004 to 9032. Any string headed with a A^C will have it removed before storage.
9119-9135	This routine takes an integer string parameter from a command and returns its numeric value in P.
9136	Program starts here.
9136-9150	Determination of file status, and building of file if it is new.
9151-9153	One time initialization
9154-9165	The new C.D. is located here with the files opened properly for editing. If an EOF is encountered, then the C.D. is set to the first element of the file.
9166-9187	Individual editing commands are put into A$, and the ' command is decoded. Spaces are also removed from all commands except SSUB and SF.

STATEMENT RANGE	DESCRIPTION
9188-9193	.M. command is processed
9194-9196	.T. command is processed
9197	The sequential location of the C.D. is saved.
9198-9199	Trap for instruction trace.
9200-9215	I, R, & D commands are processed.
9216-9233	Subroutine to list C.D. P9 controls formatting.
9234-9267	C family of commands is processed.
9268-9272	MG command is processed.
9273-9304	L command is executed.
9305-9311	Routine to write C.D. onto file F2
9312-9335	NF & SF commands executed.
9336-9346	PS command executed.
9347-9359	I/O routines for F1 & F2, used by the PS, PR, and J commands.
9360-9377	PR command done here.
9378-9449	A and A sub-group commands executed here.
9450-9455	E command processed, and exit.
9456-9464	PF command done here.
9465-9467	Trap for unrecognized command.
9468-9491	NSUB command processed.
9492-9523	SSUB command processed.
9524-9573	J command processed.
9574-9582	Subroutine to get parameters for substitution commands.
9583-9594	DEF executed.
9595-9606	MAC executed.
9999	End of program

RUN

```
GET-MACRO
10 FILES TEST,SCR
RUN
MACRO

OLD OR NEW FILE?OLD
>?L $E

THIS*IS*THE*FIRST*LINE*OF*A*SAMPLE*FILE*FOR*THE*PROGRAM*LIBRARY.
AND*THIS*IS*THE*SECOND*LINE.
FINALLY,*THIS*IS*THE*THIRD*AND*LAST*LINE.
>?DEF 1
$?SSUB \*MAC 1
>?MAC 1
>?J 30
>?MG 20
>?L $E
                    THIS IS THE FIRST LINE OF A
                    SAMPLE FILE FOR THE PROGRAM
                    LIBRARY. AND THIS IS THE
                    SECOND LINE. FINALLY, THIS IS
                    THE THIRD AND LAST LINE.
```

```
>?A
THIS IS THE FIRST LINE OF A
*?SR
              ↑
*?I::::::::::::
*?C
THIS IS THE FIR::::::::::::ST LINE OF A
*?J 30
 'S','C','P', OR 'E' COMMAND EXPECTED
*?E
>?J 30
>?L $E
                    THIS IS THE FIR::::::::::::ST
                    LINE OF A SAMPLE FILE FOR THE
                    PROGRAM LIBRARY. AND THIS IS
                    THE SECOND LINE. FINALLY,
                    THIS IS THE THIRD AND LAST
                    LINE.
>?E

DONE
```

TITLE: INTERTERMINAL COMMUNICATOR

MESSAG
36284

DESCRIPTION:

This program allows messages to be entered at one user terminal and to be received anytime afterward at another user terminal. Messages are self-dating and include the sender's and receiver's names. It is useful, for example, when a number of schools use the same computer, as it provides a fast and simple means of sending printed messages from school to school.

INSTRUCTIONS:

Open a file named MSFILE in a semi-privileged user code (i.e., one that begins with the letter A) to which all users who are to use the program have access. The program contains buffer handling routines to prevent any confusion when two or more users are writing on the file at the same time. The size of the file is variable from 2 to 128 records. Suggested starting sizes:

For 256-word records (i.e., 2000C or 2000C'/F): 20 records.
For 64-word records (i.e., 2000B): 60 records.

The program compensates for varying number of records and varying record sizes.

The program has four options selected by number:

STOP (option number 0): Stops the program.

RECEIVE MESSAGE (option number 1): The user selects the message to be printed by its number (a listing of available messages is given at the beginning of the RUN and when option number 3 is selected). After the message is printed the user may have the option of deleting the message from the file. (The program will not allow any user except the sender of the message to remove it from the file within five minutes after it is entered. This prevents one user from deleting a message being entered concurrently by another user.)

ENTER MESSAGE (option number 2): The user inputs the receiver's and his names. The program assigns the message a number and labels it with the current data and time. The user inputs his message using as many lines as needed and types the word END for the last input to stop.

RECEIVE LIST OF AVAILABLE MESSAGES (option number 3): This option gives a listing of currently available messages. It is automatically selected at the beginning of the RUN.

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

RUN

```
RUN
STOP
RUN
MESSAG

INTERTERMINAL COMMUNICATOR


THERE ARE MESSAGES FOR:

   1. BOB COLLINS
   2. EVERYONE
   3. J BENTLY S.H.S
   4. ANYBODY


OPTION?  0 = STOP, 1 = RECEIVE MESSAGE, 2 = ENTER MESSAGE,
3 = RECEIVE LIST OF AVAILABLE MESSAGES - ?1

TYPE THE NUMBER OF THE MESSAGE YOU WANT. - ?1


FOR: BOB COLLINS     FROM: D. MCCARTNEY     3/29/73     2:54 AM

BOB, I'VE COPIED YOUR PROGRAM FOR EXPERIMENT 15 FROM B102 AND WILL
HAVE MY STUDENTS BEGIN WORK ON IT TOMORROW.  I HAVE FOUND AN ERROR IN
EXPT14.  USING DATA OF 1,2,.54,.737 I OBTAIN 'SUBSCRIPT OUT OF BOUNDS
IN LINE 1750' BEFORE THE FOURTH ITERATION IS PRINTED.
END OF MESSAGE.  DO YOU WANT IT REMOVED FROM THE FILE?YES


OPTION?2
THIS MESSAGE IS FOR?DON MCCARTNEY
AND IS FROM?BOB COLLINS

ENTER MESSAGE (ANY NUMBER OF LINES, 3 MINUTE TIME LIMIT PER
LINE).  TYPE  END  FOR LAST LINE TO STOP INPUT.

? THANKS, DON.  I HAVE ALREADY FOUND THE ERROR IN EXPT14.  I HAVE SAVED
? THE CORRECTED VERSION IN THE B100 LIBRARY.
? END

OPTION?1

TYPE THE NUMBER OF THE MESSAGE YOU WANT. - ?2


FOR: EVERYONE     FROM: EASTERN H.S.     3/29/73     2:56 AM

ANNOUNCING A NEW LIBRARY PROGRAM:  TRIG

TRIG IS A DRILL PROGRAM OF TRIGONOMETRIC ANGLES.  EXAMPLE QUESTIONS
ARE 'WHAT IS THE COSINE OF 45 DEGREES?'  STUDENT RESPONSE IS
'SQR(2)/2' OR 'WHAT ANGLE HAS .5 AS ITS SINE?'  STUDENT MAY RESPOND
EITHER '30 DEGREES' OR '150 DEGREES'.  TO USE IT GET AND RUN
$TRIG.  INSTRUCTIONS ARE GIVEN.
END OF MESSAGE.  DO YOU WANT IT REMOVED FROM THE FILE?NO


OPTION?0

DONE
```

PI2

TITLE: INFORMATION SYSTEM 36737

DESCRIPTION:

A generalized information system which operates on a user defined data base using simple commands.

PILAB is an additional program provided with PI which allows the user to take a PI file and print mailing labels on a line printer.

INSTRUCTIONS:

The PI system is designed to be a time sharing, command oriented, information system. It allows the user to create on-line data bases for the storage and retrieval of information in a form that is meaningful to the user. It requires no prior knowledge of computers or programming. The user must first decide on the structure of his data base. Then by using a few PI commands, he can manipulate the data for his own needs.

The commands which are currently implemented are:

ASSIGN - allows the user to assign a file to the PI system. The user can thus have several data bases of different structures.

COPY - copies one PI file to another. Used for expanding the size of a PI file.

CHANGE - used for changing existing entries in the data base.

DELETE - deletes any entry by entry number.

ENTER - allows the user to enter new data into the data base.

FIND - allows the user to find any entry that meets certain conditions specified by the user.

INSERT - allows the user to insert an entry into the data base. Useful when the PI system is used as a text manipulator.

LAST - tells the user the number of his last entry and how much room is left in his file.

LIST - prints an entry, a range of entries or the entire file with appropriate entry numbers.

QUIT - returns the user to the time share system, or ends the operation of any command.

SORT - allows the user to sort his data base on any one of the fields he has defined.

Continued on following page.

SPECIAL CONSIDERATIONS:

None

ACKNOWLEDGEMENTS:

Edward J. Paige
HP, Intercontinental

INSTRUCTIONS continued

The PI system will recognize that PIDATA has not been used before and will ask the user to initialize the file. The first question to be answered is:

NO. OF RECORDS IN FILE - ?

The user will enter the number of records in PIDATA. The PI system will then ask the user to define the structure of his data base. The user can declare up to 20 fields with a total length of 70 characters. For example, if the data base were going to be used for a name and address file the user would define 5 fields as follows:

```
1  NAME,25
2  ADDRESS,25
3  CITY,10
4  STATE,5
5  ZIP,5
6  Q
```

The 'Q' for 'QUIT' in line 6 indicates that there are no more fields to be defined for this data base. The user has set aside 25 characters of information for each name, 25 for each address, 10 for each city, 5 for each state and 5 for each zip code. The user has defined 5 fields for use when entering, changing, inserting or sorting his data base.

The PI system then asks for the data base title. When this is entered the PI system has all the information it needs about this data base and it puts the user in command mode. This is indicated by a prompt symbol '=>'. When the user sees this prompt symbol on his terminal, he is expected to give one of the PI commands.

It should be noted that PI numbers each of the entries in the file. It is this line number that is used to 'CHANGE', 'DELETE' or 'INSERT' a line. If the user does give one of these commands then he should be aware that all entries after the one affected will be re-numbered. If the sort command is given, then all entries will be re-numbered and it is suggested that the user list the entire file for reference.

After the 'CHANGE', 'DELETE', or 'INSERT' commands are used the PI system will continue to request line numbers to allow you to operate on multiple lines. To return to command mode the user should type 'Q' for 'QUIT'.

The PI system started as the 'personnel information' system for HP Corporate personnel. They use PI for keeping an on-line data base of job applicants. If an opening occurs, they can search their data base of recent applicants and find those with the closest qualifications. Some of the fields they use are degree, field of specialization and years of experience. If they need, for example, a programmer with a B.S. in mathematics and three years of experience they can use PI to search their data base and find all recent applicants with those qualifications.

Another user of PI is the 'visitor and training' group at HP Intercontinental Division. They use PI to keep records on all visitors to ICON including HP people and customers. The fields they use are name, origin, title and type. For example, they can find any visitor who is a systems analyst with YHP Japan.

Since many of the existing applications required only slight modification to the program, it was decided to allow the user to define his own data structure independently of the program. This provides both the user and the programmer a great deal of flexibility. Furthermore, new features can be added to PI by simply adding a new module and reading the data base parameters out of the file.

RUN

```
:PE-PITEST,50
RUN
PI


FILE NAME - ?PITEST

NO. OF RECORDS IN FILE - ?50


DEFINE DATA BASE STRUCTURE.


NO.  NAME, LENGTH

 1 ?NAME,31
 2 ?ADDRESS,31
 3 ?CITY,3
 4 ?ZIP,5
 5 ?Q
```

Since file has not been initialized, PI asks for No. of records in the file and the data base structure and title.

```
DATA BASE TITLE - NAME AND ADDRESS
NAME AND ADDRESS


=> ENTER

NO.  1
NAME PAIGE, EDWARD J.
ADDRESS 3706 SPRUCE STREET
CITY NEW
ZIP 94560


NO.  2
NAME THOMPSON, DR. & MRS. DONALD
ADDRESS 806 DUBLIN ST.
CITY PLE
ZIP 94523


NO.  3
NAME FROBIC, SANDY
ADDRESS 18846 ARDEN CT.
CITY SJ
ZIP 95125


NO.  4
NAME Q


=> LAST

LAST ENTRY IS NUMBER  3
LAST RECORD IS  4
SPACE AVAILABLE FOR  326   ENTRIES.


=> ENTER


NO.  4
NAME STONE, EDWIN
ADDRESS 3999 PRESCOTT AVE.
CITY SAR
ZIP 95070


NO.  5
NAME TURNER, M&M R.S.
ADDRESS 2250 CHERRY CT.
CITY MIL
ZIP 95035


NO.  6
NAME NEWEY, THOMAS
ADDRESS 3378 WALNUD ST.
CITY FRE
ZIP 94538


NO.  7
NAME TAYLOR, JAMES
ADDRESS 4431 NEWEL ROAD
CITY PA
ZIP 94303


NO.  8
NAME PAYNE, THERESA
ADDRESS 999 WEBSTER DRIVE
CITY LIV
ZIP 94550
```

The "=>" is a signal to the user to enter a command.

The user enters data into his PI file.

'LAST' tells the user how much room he has left.

The user enters more data.

NO. 9
NAME SMITH, M&M ROBERT
ADDRESS 2669 WAVERLY
CITY LAF
ZIP 94549

NO. 10
NAME CLARK, WILLIAM
ADDRESS 5766 BLOSSOM LANE
CITY SJ
ZIP 95123

NO. 11
NAME JACKSON, M'-&]-M HUGH J.
ADDRESS 2336 CAMELLIA
CITY CUP
ZIP 950L4

NO. 12
NAME BROWN, M&M WILLIE
ADDRESS 333 NO. 33RD ST.
CITY SJ
ZIP 95113

NO. 13
NAME ERWIN, MRS. T.E.
ADDRESS 1555 POPPY WAY
CITY LG
ZIP 95030

NO. 14
NAME ROBINSON, M&M JOHN
ADDRESS 7731 ALMA COURT
CITY LAL
ZIP 94022

NO. 15
NAME WEAVER, E.S.
ADDRESS YALE COURT
CITY MV
ZIP 94040

NO. 16
NAME HENDERSEN, BETTY
ADDRESS 973 FOREST AVE.
CITY PA
ZIP 94301

NO. 17
NAME PRICE, LEE
ADDRESS 122 SPENCER WAY
CITY LAL
ZIP 94022

NO. 18
NAME MOORE, JIM
ADDRESS 989 PLANETREE DR.
CITY SUN
ZIP 94087

NO. 19
NAME GOODMAN, M&M D.P.
ADDRESS 348 FAWN DRIVE
CITY SJ
ZIP 95124

```
NO.  20
NAME LEWIS, M&M H
ADDRESS 545 MISSION
CITY SJ
ZIP 95128


NO.  21
NAME YOUNG, M&M ROBERT
ADDRESS 1011 OAKMONT DR
CITY CSL
ZIP 95051


NO.  22
NAME Q


=> LA

LAST ENTRY IS NUMBER  21
LAST RECORD IS  6
SPACE AVAILABLE FOR  308   ENTRIES.


=> LIST
                                   The user lists the entire file.
> ALL
 1
PAIGE, EDW ARD J.              3706 SPRUCE STREET          NEW94560
 2
THOMPSON, DR. & MRS. DONALD    806 DUBLIN ST.              PLE94523
 3
FROBIC, SANDY                  18846 ARDEN CT.             SJ 95125
 4
STONE, EDWIN                   3999 PRESCOTT AVE.          SAR95070
 5
TURNER, M&M R.S.               2250 CHERRY CT.             MIL95035
 6
NEWEY, THOMAS                  3378 WALNUT ST.             FRE94538
 7
TAYLOR, JAMES                  4431 NEWEL ROAD             PA 94303
 8
PAYNE, THERESA                 999 WEBSTER DRIVE           LIV94550
 9
SMITH, M&M ROBERT              2669 WAVERLY                LAF94549
 10
CLARK, WILLIAM                 5766 BLOSSOM LANE           SJ 95123
 11
JACKSON, M&M HUGH J.           2336 CAMELLIA               CUP950L4
 12
BROWN, M&M WILLIE              333 NO. 33RD ST.            SJ 95113
 13
ERWIN, MRS. T.E.               1555 POPPY WAY              LG 95030
 14
ROBINSON, M&M JOHN             7731 ALMA COURT             LAL94022
 15
WEAVER, E.S.                   YALE COURT                  MV 94040
 16
HENDERSEN, BETTY               973 FOREST AVE.             PA 94301
 17
PRICE, LEE                     122 SPENCER WAY             LAL94022
 18
MOORE, JIM                     989 PLANETREE DR.           SUN94087
 19
GOODMAN, M&M D.P.              348 FAWN DRIVE              SJ 95124
 20
LEWIS, M&M H                   545 MISSION                 SJ 95128
 21
YOUNG, M&M ROBERT              1011 OAKMONT DR             CSL95051

> QUIT


=> CHANGE
```

```
LINE NO. - 21
NAME YOUNG, M&M ROBERT                -
ADDRESS 1011 OAKMONT DR                  -
CITY CSL - SCL
ZIP 95051 -                      The user changes selected fields within
                                 certain entries.
LINE NO. - 5
NAME TURNER, M&M R.S.                 -
ADDRESS 2250 CHERRY CT.                  -
CITY MIL - PA
ZIP 95035 -

LINE NO. - 13
NAME ERWIN, MRS. T.E.                 - IRWIN, MRS. T.E.
ADDRESS 1555 POPPY WAY                   -
CITY LG  -
ZIP 95030 -

LINE NO. - QUIT


=> LIST

> 5                                          The user lists the changed
 5                                           entries to check them.
TURNER, M&M R.S.              2250 CHERRY CT.                PA 95035

> 13
 13
IRWIN, MRS. T.E.              1555 POPPY WAY                 LG 95030

> 21
 21
YOUNG, M&M ROBERT             1011 OAKMONT DR                SCL95051

> QUIT


=> DELETE

LINE NO. - 18                 The user deletes one entry.

LINE NO. - QUIT


=> LIST

> ALL                                  The user lists his file.
 1
PAIGE, EDWARD J.              3706 SPRUCE STREET             NEW94560
 2
THOMPSON, DR. & MRS. DONALD   806 DUBLIN ST.                PLE94523
 3
FROBIC, SANDY                 18846 ARDEN CT.                SJ 95125
 4
STONE, EDWIN                  3999 PRESCOTT AVE.             SAR95070
 5
TURNER, M&M R.S.              2250 CHERRY CT.                PA 95035
 6
NEWEY, THOMAS                 3378 WALNUT ST.                FRE94538
 7
TAYLOR, JAMES                 4431 NEWEL ROAD                PA 94303
 8
PAYNE, THERESA                999 WEBSTER DRIVE              LIV94550
 9
SMITH, M&M ROBERT             2669 WAVERLY                   LAF94549
 10
CLARK, WILLIAM                5766 BLOSSOM LANE              SJ 95123
 11
JACKSON, M&M HUGH J.          2336 CAMELLIA                  CUP950L4
 12
BROWN, M&M WILLIE             333 NO. 33RD ST.               SJ 95113
 13
IRWIN, MRS. T.E.              1555 POPPY WAY                 LG 95030
 14
ROBINSON, M&M JOHN            7731 ALMA COURT                LAL94022
 15
WEAVER, E.S.                  YALE COURT                     MV 94040
```

```
 16
HENDERSEN, BETTY                973 FOREST AVE.                 PA 94301
 17
PRICE, LEE                      122 SPENCER WAY                 LAL94022
 18
GOODMAN, M&M D.P.               348 FAWN DRIVE                  SJ 95124
 19
LEWIS, M&M H                    545 MISSION                     SJ 95128
 20
YOUNG, M&M ROBERT               1011 OAKMONT DR                 SCL95051

> QUIT


=> FIND

NAME PRICE, LEE
ADDRESS
CITY
ZIP

  17
PRICE, LEE                      122 SPENCER WAY                 LAL94022


=> FIND

NAME
ADDRESS
CITY SCL
ZIP

  20
YOUNG, M&M ROBERT               1011 OAKMONT DR                 SCL95051


=> FIND

NAME
ADDRESS
CITY
ZIP 94040

  15
WEAVER, E.S.                    YALE COURT                      MV 94040


=> SORT

YOU MAY SORT ON ANY ONE FIELD.
PLEASE ANSWER YES OR NO.

NAME - ?NO
ADDRESS - ?NO
CITY - ?NO
ZIP - ?YES

SORT IN PROGRESS.


SORT COMPLETED.


=> LIST

> ALL
 1
ROBINSON, M&M JOHN              7731 ALMA COURT                 LAL94022
 2
PRICE, LEE                      122 SPENCER WAY                 LAL94022
 3
WEAVER, E.S.                    YALE COURT                      MV 94040
 4
HENDERSEN, BETTY                973 FOREST AVE.                 PA 94301
 5
TAYLOR, JAMES                   4431 NEWEL ROAD                 PA 94303
```

'FIND' command is used to find entries by field.

'SORT' allows user to sort his data base on any field.

List the file which is now sorted by zip code.

```
 6
THOMPSON, DR. & MRS. DONALD     806 DUBLIN ST.                PLE94523
 7
NEWEY, THOMAS                   3378 WALNUT ST.               FRE94538
 8
SMITH, M&M ROBERT               2669 WAVERLY                  LAF94549
 9
PAYNE, THERESA                  999 WEBSTER DRIVE             LIV94550
 10
PAIGE, EDWARD J.                3706 SPRUCE STREET            NEW94560
 11
IRWIN, MRS. T.E.                1555 POPPY WAY                LG 95030
 12
TURNER, M&M R.S.                2250 CHERRY CT.               PA 95035
 13
YOUNG, M&M ROBERT               1011 OAKMONT DR               SCL95051
 14
STONE, EDWIN                    3999 PRESCOTT AVE.            SAR95070
 15
JACKSON, M&M HUGH J.            2336 CAMELLIA                 CUP950L4
 16
BROWN, M&M WILLIE               333 NO. 33RD ST.              SJ 95113
 17
CLARK, WILLIAM                  5766 BLOSSOM LANE             SJ 95123
 18
GOODMAN, M&M D.P.               348 FAWN DRIVE                SJ 95124
 19
FROBIC, SANDY                   18846 ARDEN CT.               SJ 95125
 20
LEWIS, M&M H                    545 MISSION                   SJ 95128

> QUIT


=> LAST

LAST ENTRY IS NUMBER  20
LAST RECORD IS  6
SPACE AVAILABLE FOR  309   ENTRIES.


=> QUIT

DONE
```

TITLE:

PLOTS A GIVEN FUNCTION ON THE TELETYPE

PLOT
36104

DESCRIPTION:

A program to plot a given function on the terminal. It checks for minumum and maximum Y values over the domain, excluding the undefined points, calculates the Y axis spacing, and plots the function.

INSTRUCTIONS:

Define the function in line 8900 by:

DEF FNF(X) = ...

Example: 8900 DEF FNF(X) = 25*COS(X)*SIN(X↑2/2)/(X↑2+1)

Type RUN and the program will request the following information:

1. left X end point
2. right X end point
3. desired X increment
4. points on the X-axis for which the function is undefined (for example, the denominator becomes zero)

The output will give:

1. minimum Y
2. maximum Y
3. Y-axis spacing
4. the plot with the Y-axis horizontal and the X-axis vertical on the paper.
5. allows either the X or Y values to be printed.

SPECIAL CONSIDERATIONS:

The program will not handle functions where Y is a constant over the entire range.

Error halts and messages:

"THIS IS THE CONSTANT FUNCTION" - The value of the function is constant over its range and the program cannot plot it.

"DIVIDE BY ZERO ..." -- The function has an undefined point which was not indicated in the input.

ACKNOWLEDGEMENTS:

Ted Park
Pacific Union College

RUN

```
8900 DEF FNF(X) = SIN(X)
RUN
PLOT

INPUT XMIN,XMAX,XSTEP?0,7,.4

INPUT NUMBER OF UNDEFINED POINTS?0

WOULD YOU LIKE 'X' OR 'Y' VALUES PRINTED?X

WHICH SYMBOL FOR GRAPH?*

YMIN = -.996165
YMAX =  .999574
YSTEP =  3.56382E-02

 0            YYYYYYYYYYYYYYYYYYYYYYYYYYY*YYYYYYYYYYYYYYYYYYYYYYYYYYY
 .4                                      X          *
 .8                                      X                   *
 1.2                                     X                        *
 1.6                                     X                          *
 2                                       X                       *
 2.4                                     X                   *
 2.8                                     X         *
 3.2                                   * X
 3.6                         *           X
 4                     *                 X
 4.4             *                       X
 4.8            *                        X
 5.2               *                     X
 5.6                       *             X
 6.                              *       X
 6.4                                     X  *
 6.8                                     X            *


MORE?YES
INPUT XMIN,XMAX,XSTEP?0,7,.4

INPUT NUMBER OF UNDEFINED POINTS?0

WOULD YOU LIKE 'X' OR 'Y' VALUES PRINTED?Y

WHICH SYMBOL FOR GRAPH?*

YMIN = -.996165
YMAX =  .999574
YSTEP =  3.56382E-02

 0            YYYYYYYYYYYYYYYYYYYYYYYYYYY*YYYYYYYYYYYYYYYYYYYYYYYYYYY
 .389418                                 X          *
 .717356                                 X                   *
 .932039                                 X                        *
 .999574                                 X                          *
 .909297                                 X                       *
 .675463                                 X                   *
 .334988                                 X         *
-5.83749E-02                           * X
-.442521                        *        X
-.756803                  *              X
-.951602             *                   X
-.996165            *                    X
-.883455               *                 X
-.631267                     *            X
-.279417                           *      X
 .116548                                 X  *
 .494111                                 X            *
```

TITLE:

KEYBOARD ENTRY MULTIPLE FUNCTION PLOTTER

PLOT33
36659

DESCRIPTION:

This program package (which consists of 2 program, PLOT33 and PLOTTY) inputs up to 20 functions from the keyboard; then compiles and plots them. They may be plotted on a Digilog Model 33 using cursor controls, on a teletype matrix plotter such that the graph comes out in a square (47 x 72 characters), or on a plotter that turns graphs out sideways to any length.

The axes are printed using "+"'s and integer values from -9 to 9 are shown on each axis (negative signs omitted).

The range and domain are specified by the user each time he graphs the functions.

INSTRUCTIONS:

When the computer prints "Y=" the user enters the function. Then the character that the function is to be graphed with is entered.

If you are using a Digilog Model 33 the computer will ask you if you want the graph to be horizontal or vertical. If you type "H" (for horizontal), you will get the matrix plotter. If you type "V", you will get the sideways plotter.

Next the computer asks for the x- and y- limits. Each requires two numbers; the lower and higher limits of the range and domain. Always input the lower ones first to get a proper graph. Next the computer will tell you the x- and y- spacing. They are the distances between adjacent "+"'s on each axis.

SPECIAL CONSIDERATIONS:

All functions should be in X. Input them as you would state them in a LET statement, without the "Y=." MIN, MAX, and relational operators are not acceptable. The following pre-determined functions are acceptable: SIN, COS, TAN, ATN, LOG, EXP, SQR, ABS, INT, SGN, and RND. Operations are executed in the following order: (executed first) Unary minus (-X, but not 3-X), Exponentiation, Multiplication and Division. (Executed last) Addition and Subtraction. Note that -X 2=(-X) 2, but 3-X 2=3-(X 2).

FOR INSTRUCTIONAL PURPOSES
Suitable Course(S): High school mathematics beyond elementary algebra.

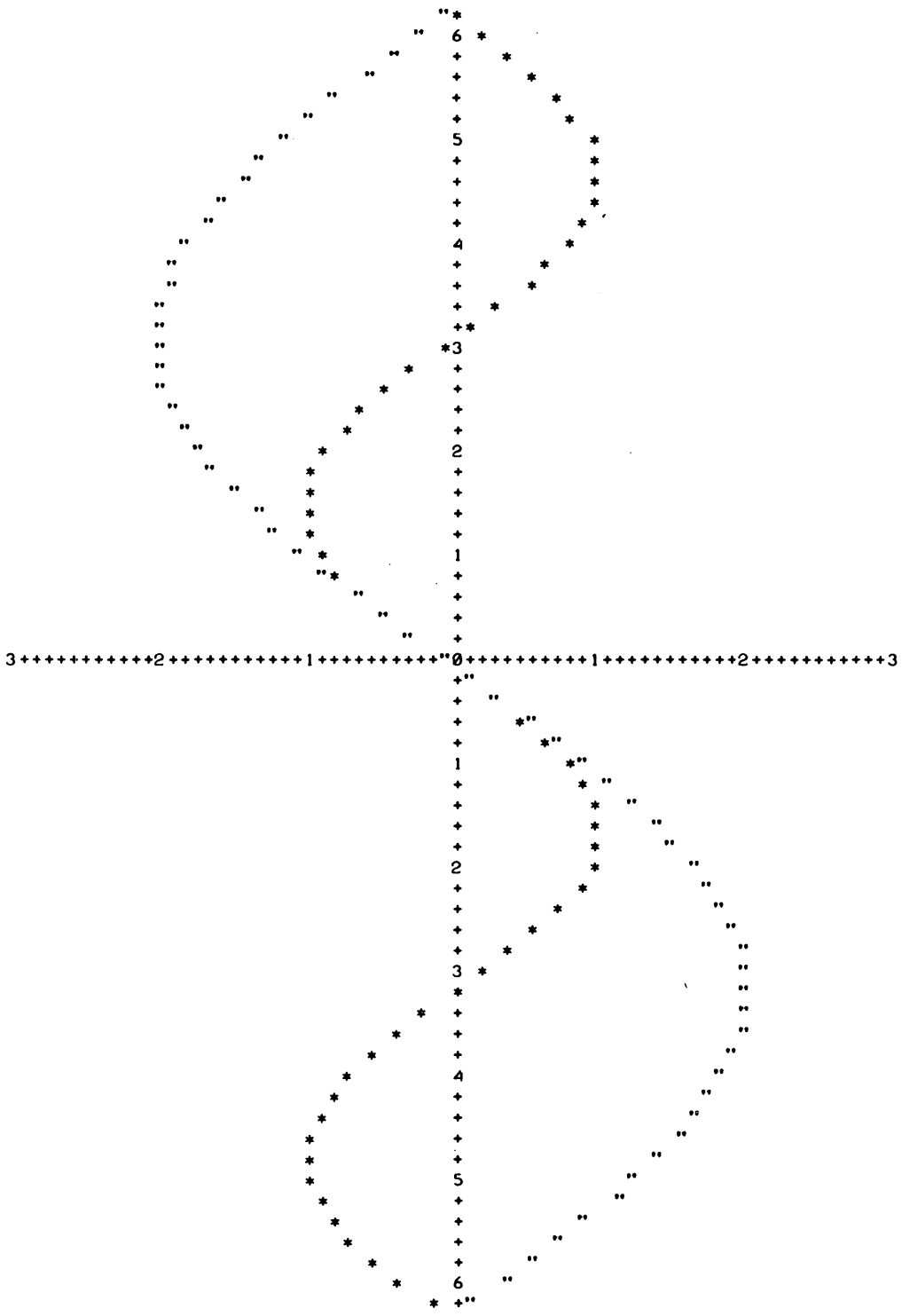
Student Background Required: Elementary algebra

Students may study the shapes of graphs of different functions and how they are changed by changing the coefficients.

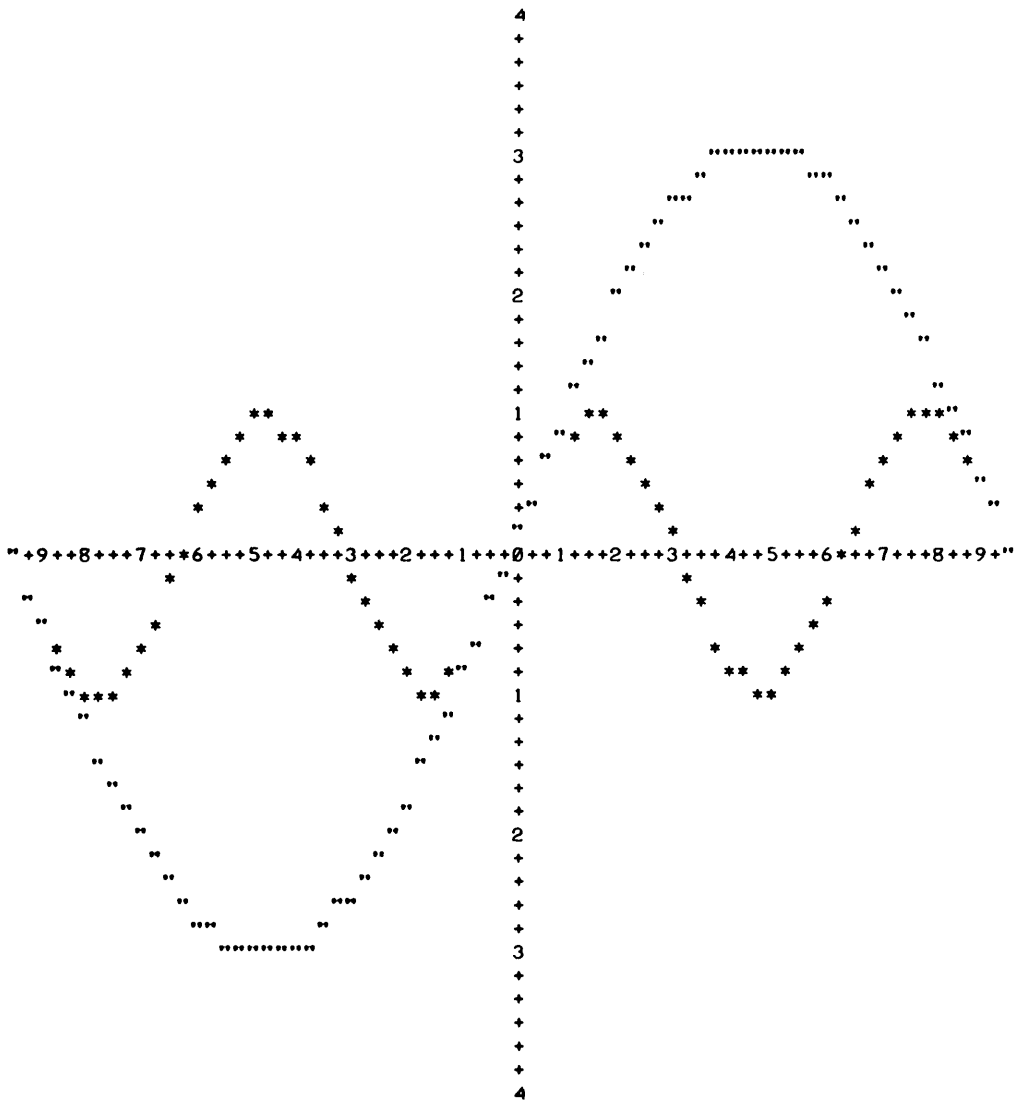
ACKNOWLEDGEMENTS:

Paul Vojta
Southwest High School

```
RUN

RUN
PLOT33

Y=SIN(X)
WHAT WILL BE THE PRINTED CHARACTER? *
IS THERE ANOTHER FUNCTION TO BE PLOTTED?YES
Y=SIN(X/C)*C
WHAT WILL BE THE PRINTED CHARACTER? "
IS THERE ANOTHER FUNCTION TO BE PLOTTED?N
ARE YOU USING A DIGILOG 33?N
X LIMITS?-6.283,6.283
Y LIMITS?-3,3
DO YOU WANT THE GRAPH TO BE HORIZONTAL OR VERTICAL?V
Y-SPACING= .084507
X-SPACING?.2
CONSTANT DATA:
C=?2
```




```
AGAIN?Y
X LIMITS?-9.4248,9.4248
Y LIMITS?-4,4
DO YOU WANT THE GRAPH TO BE HORIZONTAL OR VERTICAL?H
X-SPACING= .265487
Y-SPACING= .173913
CONSTANT DATA:
C=?3
```



AGAIN?NO

DONE

CONTRIBUTED PROGRAM **BASIC**

TITLE:

ASCII CHARACTER PLOTTER FOR 7200 PLOTTER

PLOTS
36840

DESCRIPTION:

Plots any of the 96 ASCII characters on the 7200 plotter. User specifies character, character size, X and Y coordinates. Subroutine uses the character file LETRS.

INSTRUCTIONS:

GOSUB 9000 with the following variables set:

Input:

X1 = X-coordinate of lower left point of character.
Y1 = Y-coordinate of lower left point of character.
Z1 = Size of character in terms of absolute grid points (square).
Z2 = Character number.
Z3 = File # of character file (LETRS).

Inside:

Z4,Z5,Z6,Z7,Z8

Returns:

Z9 = 1, successful
Ø, unsuccessful

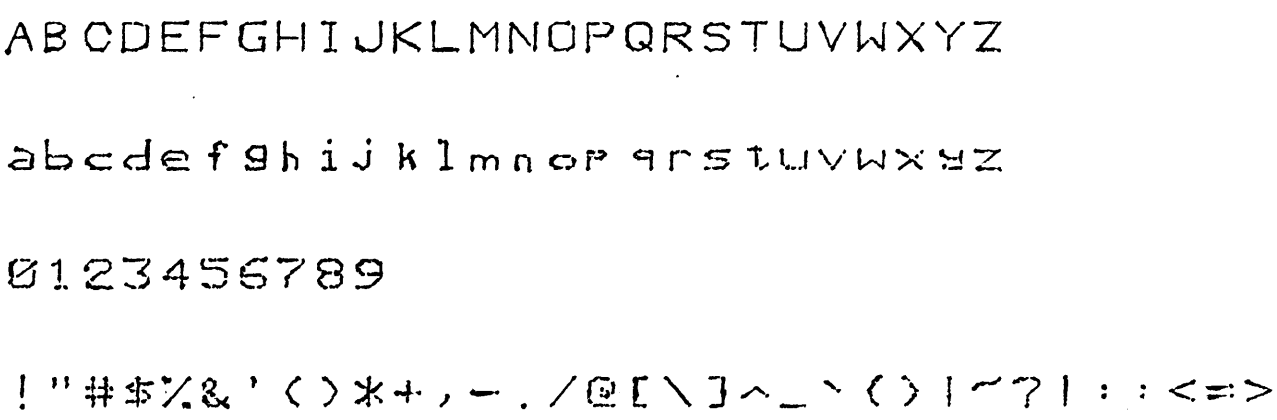
Character Numbers

A-1	T-2Ø	c-39	v-58	/-77
B-2	U-21	d-4Ø	w-59	@-78
C-3	V-22	e-41	x-6Ø	{-79
D-4	W-23	f-42	y-61	\-8Ø
E-5	X-24	g-43	z-62	}-81
F-6	Y-25	h-44	!-63	^-82
G-7	Z-26	i-45	"-64	_-83
H-8	Ø-27	j-46	#-65	`-84
I-9	1-28	k-47	$-66	{-85
J-1Ø	2-29	l-48	%-67	}-86
K-11	3-3Ø	m-49	&-68	¦-87
L-12	4-31	n-5Ø	'-69	~-88
M-13	5-32	o-51	(-7Ø	?-89
N-14	6-33	p-52	)-71	|-9Ø
O-15	7-34	q-53	*-72	:-91
P-16	8-35	r-54	+-73	;-92
Q-17	9-36	s-55	,-74	<-93
R-18	a-37	t-56	--75	=-94
S-19	b-38	u-57	.-76	>-95

ACKNOWLEDGEMENTS:

Ken Klingman
HP, Data Systems Division

RUN


ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789
!"#$%&'()*+,-./@[\]^_`(){~?|::<=>

CONTRIBUTED PROGRAM **BASIC**

PLOTWD
36228

TITLE: WORD PLOTTER

DESCRIPTION:

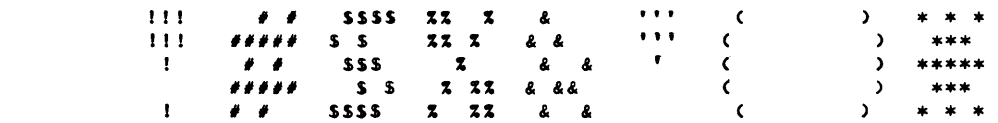
This program converts a word, which is input by the user, into a large format. Each printable character except the quote, backarrow and space can be converted into a five-by-five character format and printed. To analyze the complete character set, a user can type in the word as Z^C. The main use of this program is probably the making of chart-sized titles which can be read at a distance.

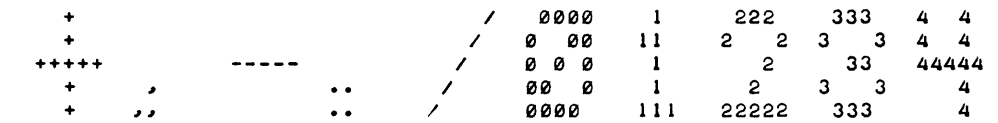
INSTRUCTIONS:

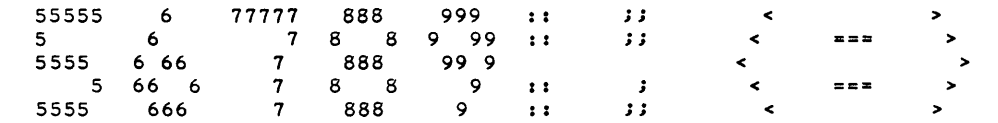
Enter any word or words (maximum 72 characters) with only printable characters or the space, but not a " or ←. If you input Z^C, the complete character set will be typed out. Answer "GO AGAIN" with either YES or NO.

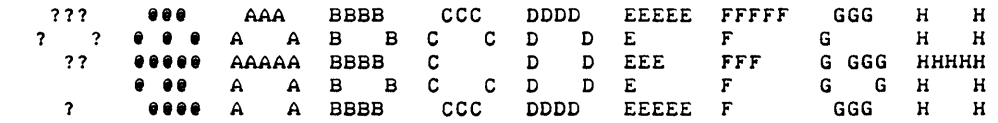
ACKNOWLEDGEMENTS: Nelson Crowle
Cherry Creek High School

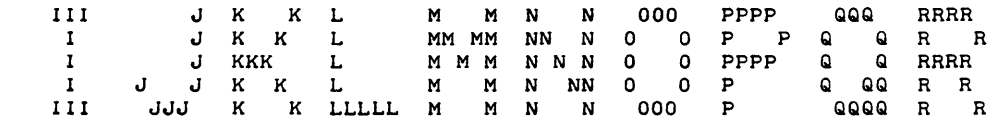
```
RUN

RUN
PLOTWD

INPUT WORD?
```

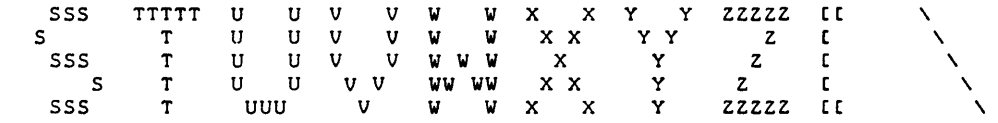












TITLE:

TWO VARIABLE PLOT PROGRAM

PLOTXY
36888-18034

DESCRIPTION:

Plots one arbitrary array against another. Data to be plotted are generated or read by a user written program and passed to PLOTXY through a COM statement. Multiple copies of output may be generated and scales may be changed.

INSTRUCTIONS:

LIMITATIONS: Plots a maximum of 60 pairs of points.

BASIC PROGRAM OPERATION: The program does not generate or read data; the data to be plotted must be provided by a user written BASIC program which must conform to the following conventions:

(1) The first statement in the program should be:

```
10 COM X (61), Y (60)
```

(2) The 61st position in the X array should contain the number of points to be plotted (60 max).

(3) X (1), ..., X (n) should contain the values of the independent variable, and Y (1), ..., Y (n) should contain the values of the dependent variable. The sample problem shows a program to generate values of the standard normal probability distribution

$$Y = \frac{1}{\sqrt{2\pi}} \quad e^{-x^2/2}$$

for values of x from -3 to 2.9 in increments of .1. There are a total of 60 points generated.

Once the data generation program is entered and run, type:

```
GET - PLOTXY
```

Note: Do not scratch the data generation program before loading PLOTXY.

When the system responds with a carriage return, type RUN.

The program will plot the X and Y values generated or read by the user's program on a grid that is 40 print positions for X and 60 print positions for Y. Points on the graph are plotted as asterisks (*); if two points coincide, a number 2 is printed instead of an asterisk; if three points coincide, a number 3 is printed; if more than three points coincide, a plus sign (+) is printed. Once the graph is completed, the number of observations and the minimum and maximum values for X and Y are printed. The program will then ask: DO YOU WANT ANOTHER COPY, (1 = YES, 0 = NO)?

SPECIAL CONSIDERATIONS:

Data to be plotted must be passed to PLOTXY through a COM statement of the form:

```
1 CØM X(61), Y(60)
```

The number of points to be plotted must be stored in the 61'st position of the X array.

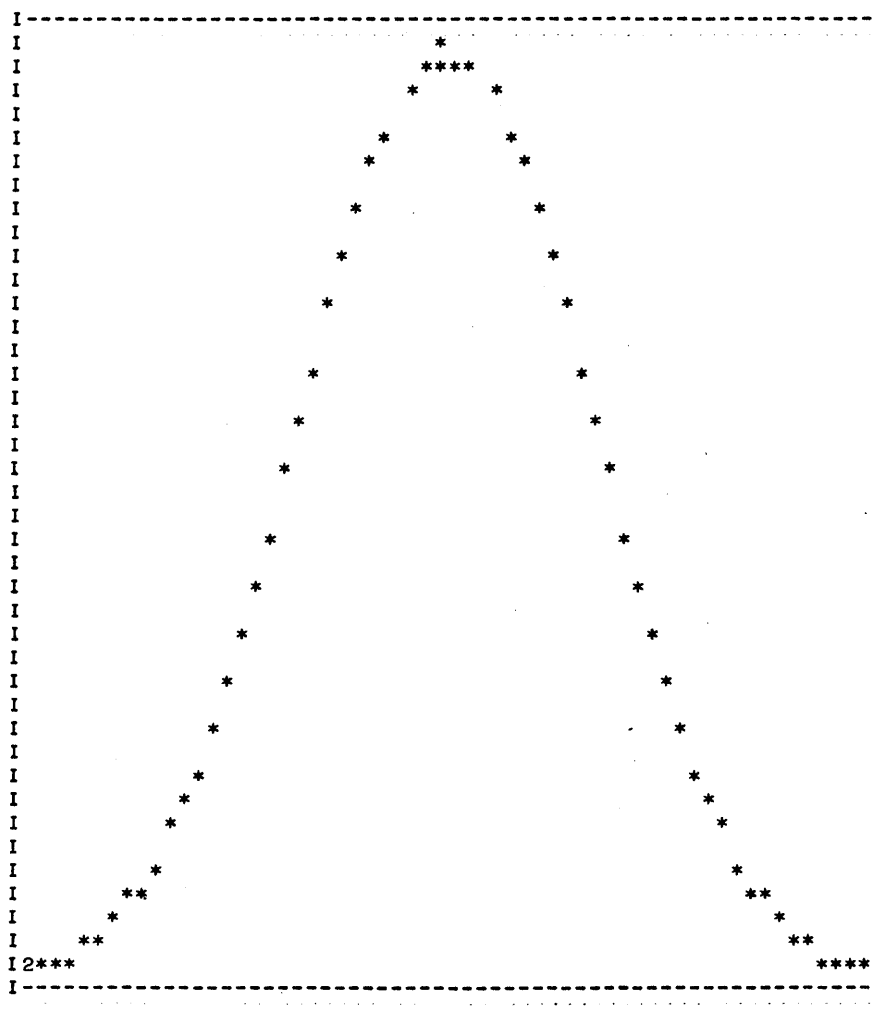
ACKNOWLEDGEMENTS:

Bill Jarosz
De Paul University

RUN

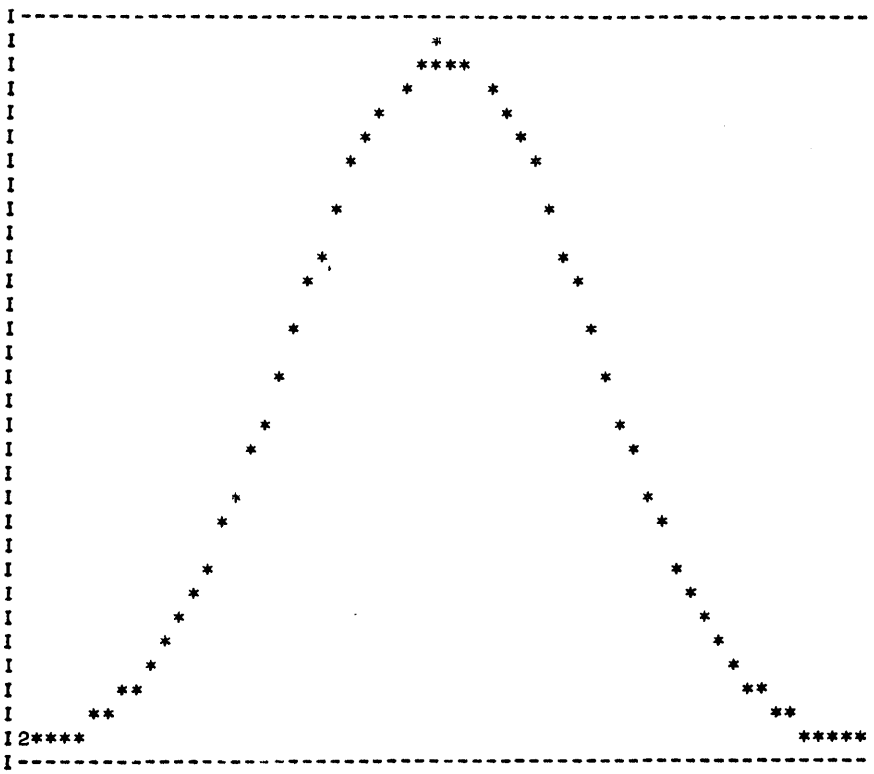
```
10  COM X[61],Y[60]
20  LET X[61]=60
30  LET Z=-3
40  LET C=1/SQR(2*3.14159)
50  FOR I=1 TO 60
60  LET X[I]=Z
70  LET Y[I]=C*EXP(-Z↑2/2)
80  LET Z=Z+.1
90  NEXT I
95  CHAIN "$PLOTXY"
100  END
```

```
RUN
BJ
```



```
NO.OBS. 60
MIN. X=-3
MAX. X= 2.9
MIN. Y= 4.43185E-03
MAX. Y= .398943
```

```
DO YOU WANT ANOTHER COPY,(1=YES,0=NO)?1

NO OF LINES FOR Y-AXIS,(40 FOR SAME SCALE AS FIRST GRAPH)?30
```




```
NO.OBS. 60
MIN. X=-3
MAX. X= 2.9
MIN. Y= 4.43185E-03
MAX. Y= .398943

DO YOU WANT ANOTHER COPY,(1=YES,0=NO)?0

DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE: GENERATES LARGE LETTERS

PRINT
36299

DESCRIPTION:

This program prints large block letters. Four different sizes may be selected (2, 3.5, 5, or 7-inch characters).

INSTRUCTIONS:

Type a '1' to the question 'INFORMATION' to receive instructions. When the machine types 'SIZE' type a number from 1 to 4 for characters of the above sizes. When the machine types 'WHAT IS YOUR MESSAGE' input a message from 1 to 72 characters. For quotes use a control K. When the machine types 'METHOD' type 1 for characters printed in the character they represent, and 2 for characters printed in a character of your choice.

ACKNOWLEDGEMENTS: Randy Gilbertson
Stillwater Senior High School

```
RUN
PRINT

INFORMATION?1
THIS PROGRAM PRINTS CHARACTERS OF HEIGHT 2, 3.5, 5, OR 7
INCHES. WHEN THE MACHINE TYPES 'SIZE?' TYPE A NUMBER
FROM 1 TO 4 FOR CHARACTERS OF THE ABOVE SIZES. WHEN
THE MACHINE TYPES 'WHAT IS YOUR MESSAGE?' INPUT A
MESSAGE FROM 1 TO 72 CHARACTERS. FOR QUOTES USE A CONTROL
K. WHEN THE MACHINE TYPES 'METHOD?' TYPE (1) CHARACTERS
PRINTED IN THE CHARACTER THEY REPRESENT (2) PRINTED
IN A CHARACTER OF YOUR CHOICE.
SIZE?1
WHAT IS YOUR MESSAGE?
H-P
METHOD?1


HHHHHHHHHHHHHHHHHH
        HH
        HH
        HH
HHHHHHHHHHHHHHHHHH


        --
        --
        --
        --
        --


PPPPPPPPPPPPPPPPP
        PP      PP
        PP      PP
        PP      PP
        PPPPPPPPPP


DONE
RUN
PRINT

INFORMATION?0
SIZE?2
WHAT IS YOUR MESSAGE?
TSB
METHOD?2
INPUT 2 OF YOUR CHARACTERS?HP
```

```
                                    HPHP
                                    HPHP
                                    HPHP
                                    HPHP
HPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHP
HPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHP
                                    HPHP
                                    HPHP
                                    HPHP
                                    HPHP

HPHPHPHP          HPHPHPHPHPHPHPHPHPHP
HPHPHPHP          HPHPHPHPHPHPHPHPHPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHPHPHPHPHPHPHPHPHP          HPHPHPHP
HPHPHPHPHPHPHPHPHPHP          HPHPHPHP

HPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHP
HPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
HPHP              HPHP              HPHP
    HPHPHPHPHPHPHP    HPHPHPHPHPHP
    HPHPHPHPHPHP      HPHPHPHPHPHP


DONE
RUN
PRINT

INFORMATION?0
SIZE?3
WHAT IS YOUR MESSAGE?
D
METHOD?1




DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
DDDDDD                                                          DDDDDD
      DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
      DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
      DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD



DONE
RUN
PRINT
```

```
INFORMATION?0
SIZE?4
WHAT IS YOUR MESSAGE?
R
METHOD?2
INPUT 2 OF YOUR CHARACTERS?XX


XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
                                XXXXXXXX                        XXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX        XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX        XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX        XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX        XXXXXXXXXXXXXXXXXXXXXXX


DONE
```

CONTRIBUTED PROGRAM **BASIC**

PSQUAR
36249

TITLE:

PATTERN SQUARES FOR 7200A PLOTTER

DESCRIPTION:

Basic pattern of four squares shown in Figure 1 at half-scale. Input parameters for full scale would be -15, 15, -10, 10. Each succeeding square is doubled in size. Inner square is one-fourth size of outer square (5/4*1, 5/4*2, 5/4*3, 5/4*4). Linear function F(X) = M x + B used in statement #15 with appropriate changes in slope and y-intercepts as shown in loops. Loop for vertical lines (#415 to #430) does not use slope intercept form of linear function.

INSTRUCTIONS:

Load tape. Select input scale parameters. Figure 2 shows overlay of basic patterns, size reduced in half for each succeeding run. Figure 3 demonstrates curcumscribed and inscribed circles. Figures 4, 5, and 6 demonstrates partial and full transformations by means of varying input scale parameters.

SPECIAL CONSIDERATIONS:

Tape for polar plotting required for circles shown in Figure #3.
Possible uses and applications:

Art Classes -- Pattern Design
Mechanical Drawing -- Scale Variations and Techniques
Algebra -- Linear Functions
Geometry -- Geometric Transformations

ACKNOWLEDGEMENTS:

John A. Cochran
Ridgefield High School
Ridgefield, Connecticut

```
9010  DATA -10,-5,1,-10,0,2,-10,5,1
9015  DATA -10,10,2,-5,10,1,0,10,2
9020  DATA 5,10,1
9025  DT←ATA -10,10,2,-7.5,7.5,2.5,-5,5,2
9030  DATA -2.5,2.5,2.5,-10,10,5,-2.5,2.5,2.5
9035  DATA -5,5,2,-7.5,7.5,2.5,-10,10,5
9040  DATA 5,10,1,0,10,2,-5,10,3
9045  DATA -10,10,2,-10,5,3,-10,0,2
9050  DATA -10,-5,1
9055  DATA -10,10,5,-7.5,7.5,2.5,-5,5,2
9060  DATA -2.5,2.5,2.5,-10,10,5,-2.5,2.5,2.5
9065  DATA -5,5,2,-7.5,7.5,2.5,-10,10,5
9500  LET X8=9999*(X-X5)/(X6-X5)
9510  LET Y8=9999*(Y-Y5)/(Y6-Y5)
9520  IF X8>9999 OR Y8>9999 THEN 9600
9530  IF X8<0 OR Y8<0 THEN 9600
9540  IF T=0 THEN 9700
9550  PRINT 2←"PLTL"
9560  LET T=0
9570  GOTO 9700
9600  IF T=1 THEN 9800
9610  PRINT "        PLTT"
9620  PRINT "OFF SCALE"
9630  LET T=1
9640  GOTO 9800
9700  PRINT INT(X8);INT(Y8)
9800  RETURN
9999  END
```

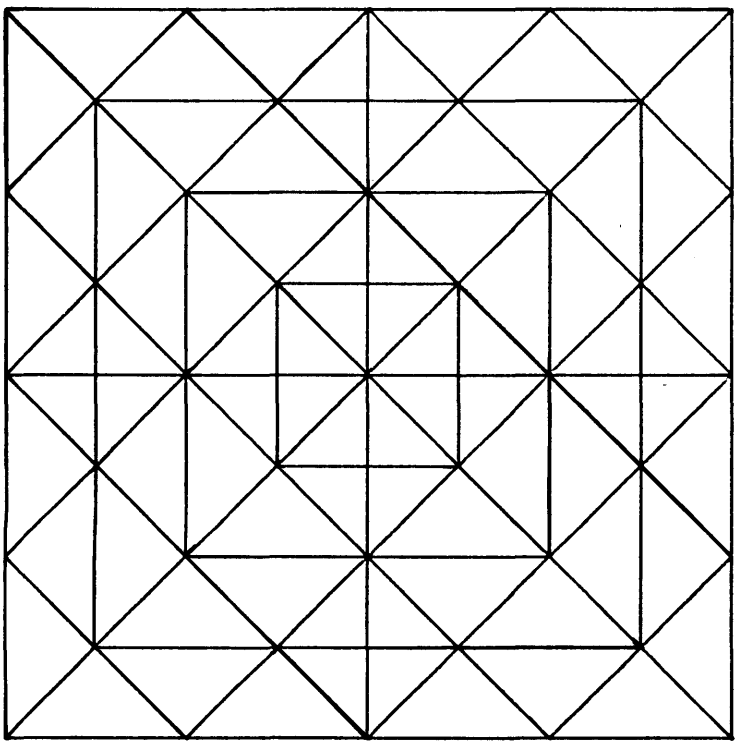


FIGURE #1

```
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-30,30,-20,20
```

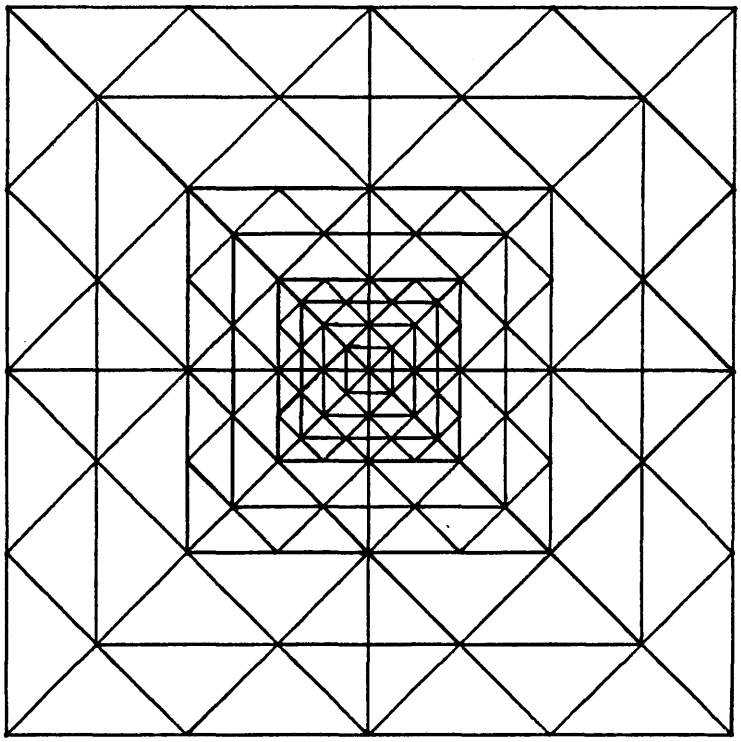


FIGURE #2

```
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-30,30,-20,20
```

```
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-60,60,-40,40
```

```
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-120,120,-80,80
```

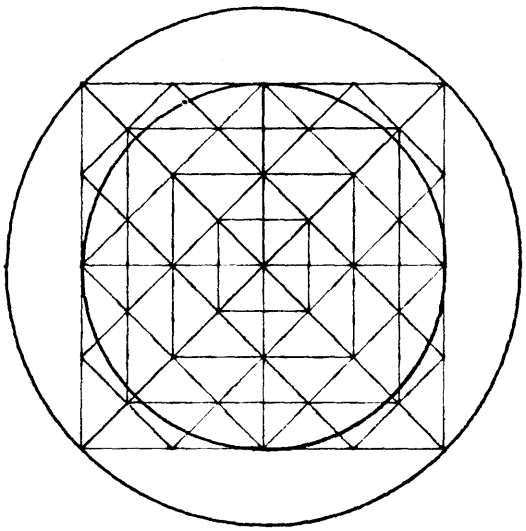


FIGURE #3

```
300  LET R=SQR (200)
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM, AND TOP OF GRAPH:
?-60,60,-40,40
INPUT INITIAL X VALUE FOR LOOP
?0
INPUT FINAL X VALUE FOR LOOP
?6.28
INPUT STEP INCREMENT
?.04
PLTL
```

```
300  LET R=10
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-60,60,-40,40
INPUT INITIAL X VALUE FOR LOOP
?0
INPUT FINAL X VALUE FOR LOOP
?6.28
INPUT STEP INCREMENT
?.04
PLTL
```

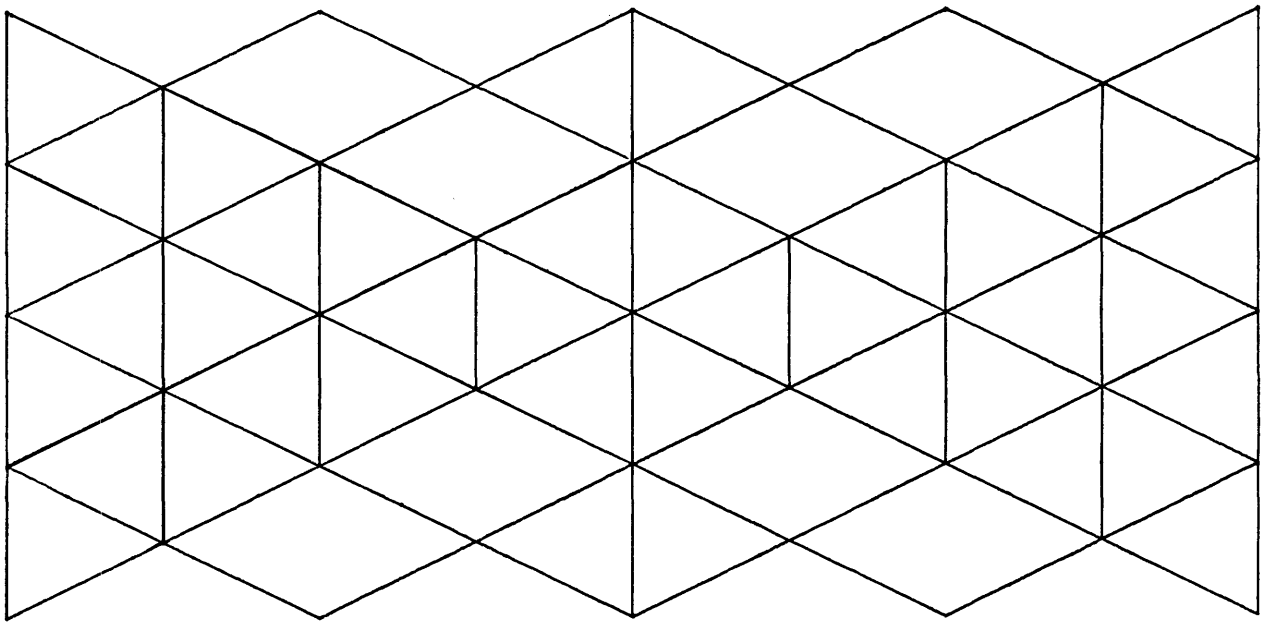


FIGURE #4

```
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-30,30,-10,10
```

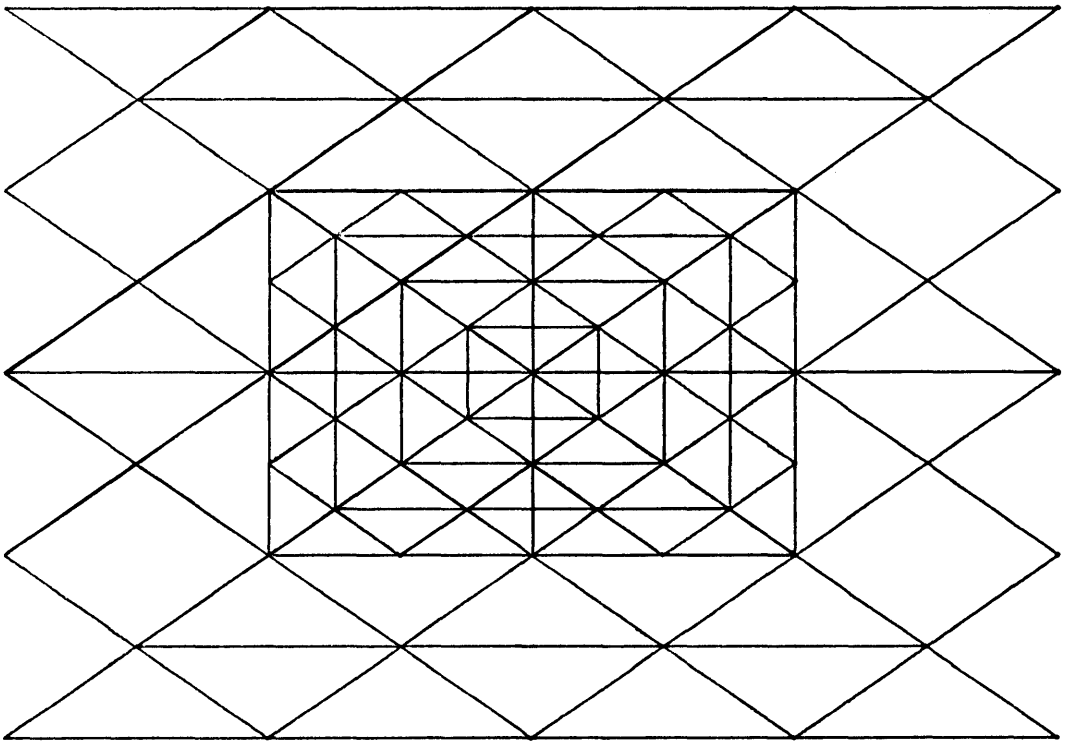


FIGURE #5

```
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-40,40,-40,40

 RUN
 INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
 ?-20,20,-20,20
```

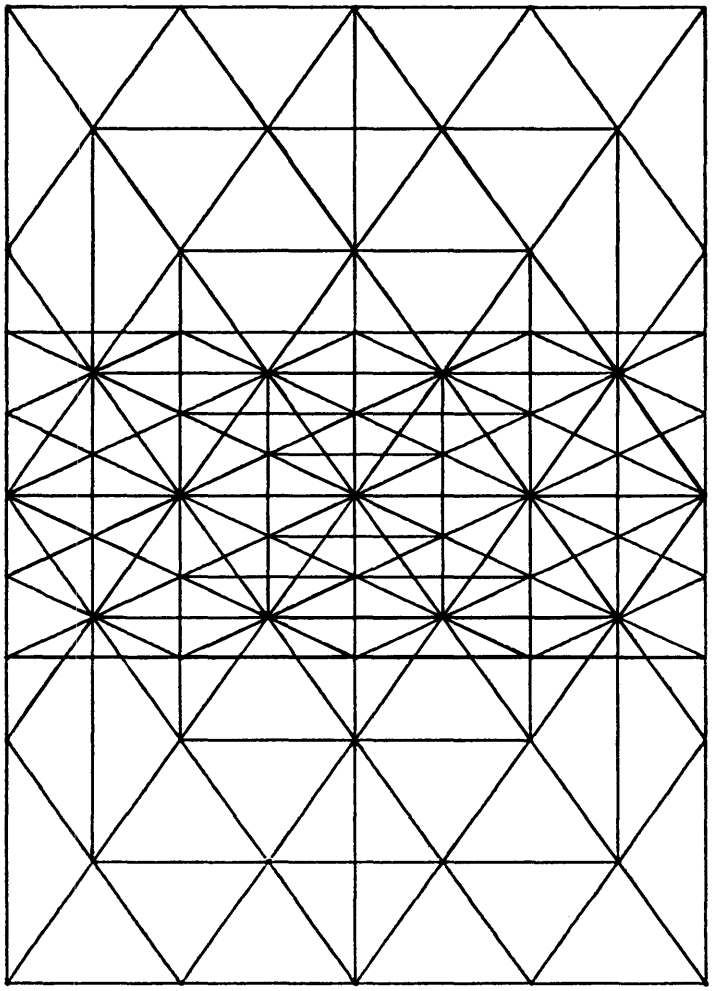


FIGURE #6

```
RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-30,30,-45,45


RUN
INPUT VALUES FOR LEFT,RIGHT,BOTTOM,AND TOP OF GRAPH:
?-30,30,-15,15
```

CONTRIBUTED PROGRAM **BASIC**

TITLE: SYSTEM LIBRARY ABSTRACTS — **SLAB 36647**

DESCRIPTION:

These two programs provide a method of giving a brief description of the system library programs. The information is stored on files by the system librarian and may be accessed by any user.

INSTRUCTIONS:

SLAB is the program to be used by the user to retrieve information. Instructions are provided during the execution of the program. The abstracts may be accessed in three ways:

1. By the name of the program ($ is optional): e.g. SLAB or $SLAB.
2. By the letter of the library (B thru Z): e.g. P would list all those programs designated as in the 'P' library. The single letter classification is arbitrary and merely provides a convenient way to subdivide the programs into 25 categories. They may correspond to the initial letter of the ID for a group of users.
3. All the abstracts may be listed by use of: A.

The program is terminated by C^C.

SLABUP is the update program and is used only by the system librarian. The commands are:

DEL	delete a given abstract
ADD	add a new entry
HEL	help gives a list of the commands
INT	initialize (must be done the first time)
LIS	list the titles of the abstracts
STO	stop execution

An entry of 'A' for library deletes the library reference. An entry of '1' for the number of programs and subprograms deletes the subprogram list.

SPECIAL CONSIDERATIONS:

The information is stored as one record per entry. Up to 192 entries may be stored (assuming 4 files, 48 records each). The record size on the HP 2000E system is 128 words. This package may be used on the HP 2000C or 2000F by limiting the size of the records when opening the files; e.g. OPE-IF1, 48, 128.

ACKNOWLEDGEMENTS: Lawrence E. Turner, Jr.
Pacific Union College

```
RUN

OPE-IF1,48
OPE-IF2,48
OPE-IF3,48
OPE-IF4,48

GET-SLABUP


RUN
SLABUP


               SYSTEM LIBRARY ABSTRACTS UPDATE


COMMAND ?HELP

COMMANDS ARE:  DEL, ADD, HEL, INT, LIS, STO

COMMAND ?ADD

ENTER NAME ?PROGRM

FILES FULL

COMMAND ?INT

ARE YOU SURE YOU WANT TO INITIALIZE?YES

INITIALIZED!!!

COMMAND ?ADD

ENTER NAME ?PROGRM


ENTER LIBRARY ?X

ENTER NUMBER OF PROGRAMS AND SUBPROGRAMS ?3

 1     PROGRM
 2    ?PROG1
 3    ?PROG2
SOURCE ?THIS IS A TEST

DESCRIPTION, '$$END$$' ENDS THE INPUT

 211
?THE DESCRIPTION GOES HERE
 183
?IT MAY BE MORE THAN ONE LINE
 153
?THE NUMBER PRECEDING THE INPUT GIVES THE NUMBER OF H-CHARACTERS LEFT
 85
?IN THE RECORD.
 69
?$$END$$


PROGRM

SUBPROGS: PROG1  PROG2

LIBRARY:  X


SOURCE:  THIS IS A TEST

THE DESCRIPTION GOES HERE
IT MAY BE MORE THAN ONE LINE
THE NUMBER PRECEDING THE INPUT GIVES THE NUMBER OF CHARACTERS LEFT
IN THE RECORD.


COMMAND ?DEL
```

```
ENTER NAME ?PROGRM


COMMAND ?ADD

ENTER NAME ?SLAB


ENTER LIBRARY ?A

ENTER NUMBER OF PROGRAMS AND SUBPROGRAMS ?2

 1      SLAB
 2     ?SLABUP
SOURCE ?L. TURNER

DESCRIPTION, '$$END$$' ENDS THE INPUT

 227
?SLAB IS A PRO\
SYSTEM LIBR\
     SYSTEM LIBRARY ABSTRACTS
 201
?SLAB IS A PROGRAM TO RETRIEVE ABSTRACTS OF SYSTEM LIBRARY PROGRAMS
 133
?STORED IN THE FILES:  IF1, IF2, IF3, AND IF4.
 85
?SLABUP IS THE UPDATE PROGRAM FOR USE BY THE SYSTEM LIBRARIAN.
 21
?$$END$$


SLAB

SUBPROGS: SLABUP

SOURCE:  L. TURNER

SYSTEM LIBRARY ABSTRACTS
SLAB IS A PROGRAM TO RETRIEVE ABSTRACTS OF SYSTEM LIBRARY PROGRAMS
STORED IN THE FILES:  IF1, IF2, IF3, AND IF4.
SLABUP IS THE UPDATE PROGRAM FOR USE BY THE SYSTEM LIBRARIAN.


COMMAND ?ADD

ENTER NAME ?ALERA


ENTER LIBRARY ?P

ENTER NUMBER OF PROGRAMS AND SUBPROGRAMS ?1

SOURCE ?L. TURNER

DESCRIPTION, '$$END$$' ENDS THE INPUT

 231
?     ALGEBRAIC ERROR ANALYSIS PROGRAM
 197
?THIS COMPUTES THE PROPAT-GATION OF ERROR THROUGH A SEQUENCE OF
 135
?BINARY ARITHMETIC OPERATIONS.
 103
?$$END$$


ALERA

LIBRARY:  P


SOURCE:  L. TURNER

ALGEBRAIC ERROR ANALYSIS PROGRAM
THIS COMPUTES THE PROPAGATION OF ERROR THROUGH A SEQUENCE OF
BINARY ARITHMETIC OPERATIONS.
```

```
COMMAND ?STOP

DONE



GET-SLAB
RUN
SLAB


               SYSTEM LIBRARY ABSTRACTS

INSTRUCTIONS ?YES


INFORMATION IS CONTAINED IN FILES CONCERNING PROGRAMS IN
THE SYSTEM LIBRARY.  THIS INFORMATION IS AVAILABLE VIA THE
PROGRAM SLAB.

THIS MAY BE ACCESSED IN THREE WAYS:
    1.  BY THE NAME OF THE PROGRAM, E.G.  SLAB OR $SLAB
    2.  BY THE LETTER OF THE LIBRARY, E.G.  P, M, OR C
    3.  A LISTING OF ALL PROGRAMS BY USING:  A


ENTER:  NAME ?SLAB


SLAB
----

SUBPROGRAMS:  SLABUP

SOURCE:  L. TURNER

SYSTEM LIBRARY ABSTRACTS
SLAB IS A PROGRAM TO RETRIEVE ABSTRACTS OF SYSTEM LIBRARY PROGRAMS
STORED IN THE FILES:  IF1, IF2, IF3, AND IF4.
SLABUP IS THE UPDATE PROGRAM FOR USE BY THE SYSTEM LIBRARIAN.


ENTER:  NAME ?ALERA


ALERA
-----

LIBRARY:  P

SOURCE:  L. TURNER

ALGEBRAIC ERROR ANALYSIS PROGRAM
THIS COMPUTES THE PROPAGATION OF ERROR THROUGH A SEQUENCE OF
BINARY ARITHMETIC OPERATIONS.


ENTER:  NAME ?
DONE
```

CONTRIBUTED PROGRAM **BASIC**

SORT
36122

TITLE: FILE SERIAL STRING SORT

DESCRIPTION:

This program will sort any serial file of strings in ascending or descending order. The sort can be made on the entire string or on any substring. No work file is required since the sort is done totally in the user file.

INSTRUCTIONS:

Simply RUN and answer the questions:

FILE?
ASCENDING OR DESCENDING ORDER?
BY SUBFIELDS (Y OR N)?
FIRST, LAST?

SPECIAL CONSIDERATIONS:

To modify for use with the 2000A or B, remove the assign statements and set up the "file" statement to have two pointers to the same file.

File must be a serial file of only strings.

ACKNOWLEDGEMENTS: George A. Tibaldi
Hewlett-Packard/Eastern Sales Region

RUN

```
RUN
SORT

FILE?A
ASCENDING OR DESCENDING ORDER?ASCENDING
BY SUB FIELDS(Y OR N)?NO
 15    ITEMS SORTED      9     PASSES

DONE
GET-FILIST
8900 FILES A
RUN
FILIST

IS T/S AN HP 2000 'A', 'B', OR 'C'?C

STOP LISTING FILE 1     AT THE FIRST EOF (Y OR N OR Q)?Y


FILE 1     RECORD 1

ARE CHANTING LOW, ARE CHANTING LOW
BRING HOME THE PATHWASYS OF TOMORROW, FROM THE SKIES,
BRING THE DREAMS OF STARS TO TIRED EYES
BRING THE WOODED SONGS TO THE CITIES
DREAM OF THE DAYS THAT PASS BEFORE US
DREAM OF THE INDIAN FIRES GLOWDREAM OF THE LAND WHERE LATIN VOICES
FOLLOW WINDING PATHS THROUGH THE FORESTS, FOLLOW GENTLE STREAMS
FOLLOW WINDING PATHS THROUGH THE FORESTS, FOLLOW GENTLE STREAMS TO
FROM THE SKIES
LAKES OF BLUE, FOLLOW THE STARS THAT SHINE AT EVEN WHEN DAY IS THROUGH
Q

FILE 1     RECORD 2

THROUGH LAKES OF BLUE, FOLLOW THE STARS THAT SHINE AT EVEN
WHEN DAY IS THROUGH           WHEN DAY IS THROUGH, WHEN DAY IS THROUGH
END OF FILE 1

STOP LISTING FILE 2     AT THE FIRST EOF (Y OR N OR Q)?Q

DONE
```

CONTRIBUTED PROGRAM **BASIC**

SPSORT
36736

TITLE:

SPEED SORT - GENERAL PURPOSE FILE SORT

DESCRIPTION:

This program sorts random access files. Records may be comprised of any series of string and numeric variables, but must be fixed format. Any field may be designated as sort key. Subfields of alpha fields may be used. Method: Splitting sort of ACM Algorithm 347.

INSTRUCTIONS:

Open input and output files, access SPSORT, and RUN.

Answer questions: input file

output file

number of field to be key for sort (first field in record is 1)

if key is alpha - specify subfield first position and last position (may be set to 1,72 if desired)

program prints file format, start time, stop time and input record count

input file is not altered.

There are three programs in this package: SPSORT, NUSORT, and ALSORT

ACKNOWLEDGEMENTS:

Bob Lewis
Hewlett-Packard/BAEDP

```
RUN

RUN
SPSORT

FILE NAMES -- INPUT?TSPFIL
           -- OUTPUT?TEST
RECORD FORMAT BY FIELD  A  A  N  A  N  A  N  A  END
SORT KEY FIELD NUMBER?4
ALPHA SORT KEY -- SUBFIELD FIRST POS, LAST POS?1,6
 498  ITEMS ON FILE
SORT START TIME 10   : 37
END OF SORT -- TIME 10   : 41
OUTPUT COMPLETED

DONE



RUN
LIST

36257A          C104            00-7302-NEW     -ASCII
ASCII CODE GENERATOR
A05Z99Z99Z99

36152A          A820            00-7107-EDU     AC-1
COMPUTER AIDED PRACTICE IN EE AC ANALYSIS
513Z99Z99Z99

36057A          A513            00-7012-BGN     ACNODE
AC CIRCUIT ANALYSIS PROGRAM
820Z99Z99Z99

36293A          F513            00-7306-NEW     ACTFIL
ACTIVE FILTER DESIGN
Z99Z99Z99Z99

36231A          A102            00-7210-NEW     ADDRES
ADDRESS LABELS
L06Z99Z99Z99

36353A          S833            00-7112-EDU     AFLIFE
JOHN CONWAY'S GAME OF LIFE
508Z99Z99Z99

36603B          E833            00-7310-EDU     ALERA
PROPAGATION OF ERROR
505Z99Z99Z99

36296B          E104            00-7310-REV     ALFTOV
ALPHA TO VARIABLE CONVERSION
105Z99Z99Z99

36292A          A107            00-7306-NEW     ALPHA
ALPHANUMERIC SORT
A05A03Z99Z99

36056A          A513            00-7012-BGN     ANALAD
CIRCUIT ANALYSIS
820Z99Z99Z99

36294A          A410            00-7306-NEW     ANCOV
ANALYSIS OF COVARIANCE
Z99Z99Z99Z99

36074A          A712            00-7012-BGN     ANNUIT
ANNUITY ANALYSIS
713Z99Z99Z99

36271A          F410            00-7306-NEW     ANOVA3
THREE FACTORIAL ANALYSIS OF VARIANCE
Z99Z99Z99Z99
```

36039B A410 00-7109-REV ANVAR1
ANALYSIS OF VARIANCE FOR A RANDOMIZED ONE-WAY DESIGN
E07Z99Z99Z99

36040B A410 00-7109-REV ANVAR2
ANALYSIS OF VARIANCE (LATIN SQUARE DESIGN)
E07Z99Z99Z99

36172A A410 00-7109-NEW ANVAR3
ANALYSIS OF VARIANCE FOR A TWO VARIABLES OF CLASSIFICATION DESIGN
E07Z99Z99Z99

36173A A410 00-7109-NEW ANVAR4
TWO-WAY ANALYSIS OF VARIANCE FOR A TWO-WAY EXPERIMENT
409Z99Z99Z99

36668A E801 00-7310-EDU AREA
COMPUTER-AUGMENTED CALCULUS TOPICS
P03Z99Z99Z99

36256B C104 00-7306-NEW ASCII*
CREATES AN ASCII FILE CONTAINING ALL 256 ASCII CHARACTERS
A05Z9999Z99

36705A E833 00-7310-EDU ATOM
DERIVES THE ELECTRONIC CONFIGURATION OF ANY ELEMENT
505507Z99Z99

36202A C720 00-7210-EDU ATTEND
ABSENTEE LISTING
Z99Z99Z99Z99

36613A A833 00-7306-HUN ATWT
CALCULATES ATOMIC WEIGHT
507Z99Z99Z99

36340A S720 00-7112-HUN AVERGI
AVERAGES AND CURVES GRADES
702408Z99Z99

36614A A833 00-7306-HUN AVOGA
AVOGADRA'S NUMBER
507Z99Z99Z99

36360A S801 00-7112-EDU BAGELS
THREE-DIGIT NUMBER GUESSING
903Z99Z99Z99

36328A S860 00-7112-HUN BALANC
TRADE AND PAYMENT BALANCES
830860Z99Z99

36075A A712 00-7012-BGN BALSHT
PROFORMA INCOME STATEMENT AND BALANCE SHEET
716Z99Z99Z99

36329B E860 00-7310-HUN BANK
SOLVES FINANCIAL PROBLEMS
801830Z99Z99

36605A F903 00-7310-NEW BASKET
BASKETBALL STATISTICS
408720Z99Z99

36116B E211 00-7310-REV BASTES
BASIC TEST PROGRAM
B01Z99Z99Z99

36105A A903 00-7012-BGN BATNUM
PLAYS THE BATTLE OF NUMBERS GAME
Z99Z99Z99Z99

36236A A903 00-7210-NEW BATTLE
BATTLESHIP GAME
801Z99Z99Z99

36109A A515 00-7012-BGN BEMDES
RECOMMENDS CORRECT STEEL BEAM USE
820Z99Z99Z99

STGINT
36176

TITLE: STRING-INTEGER CONVERSIONS

DESCRIPTION:

This subroutine will convert positive integers to string representation or strings to positive integers. String representation right-justifies characters to a specified length Z9 and fills the remaining character places with zeroes.

INSTRUCTIONS:

The main program must dimension Z$ large enough to handle the desired length of string. Z9 must be set equal to string length (actual size in the case of number to string conversion - maximum in the case of string to number).

Enter the subroutine (GOSUB 9000)

A. For string to number conversion:
   1. Set Z8=Ø
   2. Enter with Z$=string
   3. The number will be returned as "Z"
B. For number to string conversion:
   1. Set Z8=1
   2. Enter with Z=number
   3. The string will be returned as "Z$"

SPECIAL CONSIDERATIONS:

Variables used are Z, Z$, Z1-Z9 and Y$.

ACKNOWLEDGEMENTS: W. Edwin Smith
Hewlett-Packard/Data Systems

```
10 DIM Z$(6)
20 Z9=6
30 Z8=1
40 INPUT Z
50 GOSUB 9000
60 PRINT Z$
70 GO TO 40
80 STOP
APPEND-STGINT
```

RUN

```
RUN
STGINT

?123
000123
?42
000042
?37859
037859
?123456
123456
?22
000022
?689
000689
?
DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE:

SORTS STRINGS FROM FILES

STGSRT
36145

DESCRIPTION:

Using a "bucket" sort approach, STGSRT will do numeric or ASCII sorts on specified strings or sub-strings. Time required for sorting is directly proportional to number of records in the file, rather than to the square of the number, making it practical for sorting large data bases.

INSTRUCTIONS:

Create your Data Base file (DATABS) so that related data is contained within one record. The field to be sorted must be created as strings. Open a second file (DIREC) which will be written by this program as the sorted directory to the Data Base file. The structure of the example Data Base is as follows:

String	1	A$(6)	Product Number
	2	B$(4)	Classification Code
	3	C$(11)	Source, Date, Status
	4	D$(72)	Title
	5	E$(3)	Cross-Reference Code
	6	F$(3)	Cross-Reference Code
	7	G$(3)	Cross-Reference Code
	8	H$(3)	Cross-Reference Code

Dimensions and printout formats can be changed to fit various applications. Normal termination of the program will leave the most recently sort directory in file DIREC. This can then be used by other programs which need sorted entry into the Data Base.

SPECIAL CONSIDERATIONS:

By making smaller matrices and using smaller records and files, 2000A or B Systems may be used.

ACKNOWLEDGEMENTS:

W. Edwin Smith
Hewlett-Packard/Data Products Group

```
RUN

RUN
STGSRT

NUMBER OF DATA RECORDS IN FILE IS  20
WHAT IS RELATIVE POSITION OF STRING WITHIN RECORD UPON WHICH TO
SORT?  (1, 2, 3, OR 4)  ?1
WHAT ARE CHARACTER POSITIONS WITHIN STRING UPON WHICH TO SORT??
(IE  1,6   OR   2,30   ETC)  ?1,6
SPECIFY SORT TYPE AS EITHER ASCII OR NUMERIC  ?ASCII
SORT ON CHARACTER  6      IS COMPLETE
SORT ON CHARACTER  5      IS COMPLETE
SORT ON CHARACTER  4      IS COMPLETE
SORT ON CHARACTER  3      IS COMPLETE
SORT ON CHARACTER  2      IS COMPLETE
SORT ON CHARACTER  1      IS COMPLETE
PRINT DATA FILE ON CURRENT DIRECTORY ORDER? (YES OR NO) ?YES



13023A (A016)     BCS HP 7970 MAGNETIC TAPE DRIVER (D.23)
20019C (A010)     BCS CARD READER DRIVER (D.11)
20297B (A013)     RTE 2310/2311 SUBSYSTEM DRIVER (DVR56)
20307A (A009)     8K SIO TAPE PUNCH DRIVER
20334C (A016)     16K SIO HP 3030 MAGNETIC TAPE DRIVER
20339B (A216)     TEST:  2310A/B SUBSYSTEM
20533A (A105)     CONVERSION ROUTINE CON34
20581A (A014)     DOS PLOTTER DRIVER (DVR10)
20878B (A008)     2000A TO 2000B CONVERSION
20993A (A010)     DOS MARK SENSE DRIVER, KIT 12602A, (DVR15)
22065A (A018)     FORTRAN TRANSLATOR, IBM TO HP
22068A (A006)     HP 3450A DIGITAL VOLTMETER DRIVER - FORTRAN CALLABLE
22070A (A015)     HP 2773A/74A/75A DRUM DRIVER - FORTRAN CALLABLE
22108B (A006)     HP 3450A DATA SOURCE INTERFACE DRIVER - BASIC CALLABLE
22169A (A107)     ARRANGING A FLOATING POINT ARRAY
22171A (A101)     FORTRAN UNIT REFERENCE NUMBER EDITOR
22223A (A017)     LOADER BOOTSTRAP
24159A (A018)     DOS-M FORTRAN COMPILER
24166A (A011)     16K SIO HP 2767 LINE PRINTER DRIVER
24172A (A007)     BCS INPUT/OUTPUT CONTROL, BUFFERED


ADDITIONAL SORT ON MORE SIGNIFICANT STRING?  (YES OR NO) ?YES
WHAT IS RELATIVE POSITION OF STRING WITHIN RECORD UPON WHICH TO
SORT?  (1, 2, 3, OR 4)  ?2
WHAT ARE CHARACTER POSITIONS WITHIN STRING UPON WHICH TO SORT??
(IE  1,6   OR   2,30   ETC)  ?2,4
SPECIFY SORT TYPE AS EITHER ASCII OR NUMERIC  ?NUMERIC
SORT ON CHARACTER  4      IS COMPLETE
SORT ON CHARACTER  3      IS COMPLETE
SORT ON CHARACTER  2      IS COMPLETE
PRINT DATA FILE ON CURRENT DIRECTORY ORDER? (YES OR NO) ?YES



(A006) 22068A     HP 3450A DIGITAL VOLTMETER DRIVER - FORTRAN CALLABLE
(A006) 22108B     HP 3450A DATA SOURCE INTERFACE DRIVER - BASIC CALLABLE
(A007) 24172A     BCS INPUT/OUTPUT CONTROL, BUFFERED
(A008) 20878B     2000A TO 2000B CONVERSION
(A009) 20307A     8K SIO TAPE PUNCH DRIVER
(A010) 20019C     BCS CARD READER DRIVER (D.11)
(A010) 20993A     DOS MARK SENSE DRIVER, KIT 12602A, (DVR15)
(A011) 24166A     16K SIO HP 2767 LINE PRINTER DRIVER
(A013) 20297B     RTE 2310/2311 SUBSYSTEM DRIVER (DVR56)
(A014) 20581A     DOS PLOTTER DRIVER (DVR10)
(A015) 22070A     HP 2773A/74A/75A DRUM DRIVER - FORTRAN CALLABLE
(A016) 13023A     BCS HP 7970 MAGNETIC TAPE DRIVER (D.23)
(A016) 20334C     16K SIO HP 3030 MAGNETIC TAPE DRIVER
(A017) 22223A     LOADER BOOTSTRAP
(A018) 22065A     FORTRAN TRANSLATOR, IBM TO HP
(A018) 24159A     DOS-M FORTRAN COMPILER
(A101) 22171A     FORTRAN UNIT REFERENCE NUMBER EDITOR
(A105) 20533A     CONVERSION ROUTINE CON34
(A107) 22169A     ARRANGING A FLOATING POINT ARRAY
(A216) 20339B     TEST:  2310A/B SUBSYSTEM
```

```
ADDITIONAL SORT ON MORE SIGNIFICANT STRING?  (YES OR NO) ?YES
WHAT IS RELATIVE POSITION OF STRING WITHIN RECORD UPON WHICH TO
SORT?  (1, 2, 3, OR 4)  ?4
WHAT ARE CHARACTER POSITIONS WITHIN STRING UPON WHICH TO SORT??
(IE  1,6   OR   2,30   ETC)  ?1,8
SPECIFY SORT TYPE AS EITHER ASCII OR NUMERIC  ?ASCII
SORT ON CHARACTER  8       IS COMPLETE
SORT ON CHARACTER  7       IS COMPLETE
SORT ON CHARACTER  6       IS COMPLETE
SORT ON CHARACTER  5       IS COMPLETE
SORT ON CHARACTER  4       IS COMPLETE
SORT ON CHARACTER  3       IS COMPLETE
SORT ON CHARACTER  2       IS COMPLETE
SORT ON CHARACTER  1       IS COMPLETE
PRINT DATA FILE ON CURRENT DIRECTORY ORDER? (YES OR NO) ?YES



24166A  16K SIO HP 2767 LINE PRINTER DRIVER
20334C  16K SIO HP 3030 MAGNETIC TAPE DRIVER
20878B  2000A TO 2000B CONVERSION
20307A  8K SIO TAPE PUNCH DRIVER
22169A  ARRANGING A FLOATING POINT ARRAY
20019C  BCS CARD READER DRIVER (D.11)
13023A  BCS HP 7970 MAGNETIC TAPE DRIVER (D.23)
24172A  BCS INPUT/OUTPUT CONTROL, BUFFERED
20533A  CONVERSION ROUTINE CON34
20993A  DOS MARK SENSE DRIVER, KIT 12602A, (DVR15)
20581A  DOS PLOTTER DRIVER (DVR10)
24159A  DOS-M FORTRAN COMPILER
22065A  FORTRAN TRANSLATOR, IBM TO HP
22171A  FORTRAN UNIT REFERENCE NUMBER EDITOR
22070A  HP 2773A/74A/75A DRUM DRIVER - FORTRAN CALLABLE
22068A  HP 3450A DIGITAL VOLTMETER DRIVER - FORTRAN CALLABLE
22108B  HP 3450A DATA SOURCE INTERFACE DRIVER - BASIC CALLABLE
22223A  LOADER BOOTSTRAP
20297B  RTE 2310/2311 SUBSYSTEM DRIVER (DVR56)
20339B  TEST:  2310A/B SUBSYSTEM



ADDITIONAL SORT ON MORE SIGNIFICANT STRING?  (YES OR NO) ?NO

DONE
```

TITLE:

SYSDAT
SYSTEM DATE UTILITY 36634

DESCRIPTION:

SYSDAT is a utility subroutine which will provide the calling program with the current system date, the time of day, the day of the week, the user's port number, and a numeric date in any desired order. The calling program must pass its program name, the returning statement number, and the desired date format codes in that order. The calling program must begin with a COM statement in the following format:

```
10  COM N$(7),R,D$(45)
```

and must contain the following sequence:

```
20  N$="YOUR PROGRAM NAME"
30  R=RETURNING STATEMENT NUMBER (EX. 60)
40  D$="DATE FORMAT CODES" (EX. "DYMN")
50  CHAIN "$SYSDAT"
60  REM RETURN HERE WITH DATE IN R AND D$
```

SYSDAT will return with the formatted data in D$ and/or the numeric date in R. The date format requires the use of the following codes:

```
M = MONTH (JANUARY, FEBRUARY, ETC.)
D = DAY OF THE MONTH (1-31)
Y = YEAR (1972, 1973, ETC.)
W = DAY OF THE WEEK (SUNDAY, MONDAY, ETC.)
T = TIME OF DAY (5:06 PM)
P = PORT NUMBER (PORT #3)
N = NUMERIC DATE (IN YYMMDD FORMAT)
```

For example, if you pass "DMY" then you will receive 23 August 1972, or if you pass "TWMDYN" you will receive 4:07 PM Wednesday August 23, 1972 with 720823 as the numeric value in R.

INSTRUCTIONS:

The calling program must begin with the following statements:

```
10  COM N$(7),R,D$(45)
20  N$="YOUR PROGRAM NAME"
30  R=60
40  D$="DATE FORMAT CODES"
50  CHAIN "$SYSDAT"
60  REM RETURN HERE WITH DATE IN R AND D$
```

continued on following page

SPECIAL CONSIDERATIONS:

Modifications must be made to the calling program's COM statement if the calling program wishes to pass the date to another chained program.

ACKNOWLEDGEMENTS:

Dick Rands
HP, BAEDP

INSTRUCTIONS continued

For example, a program called TSTDAT might be the following:

```
     10  COM N$(7),R,D$(45)
     20  N$="TSTDAT"
     30  R=60
     40  D$="TWMDYPN"
     50  CHAIN "$SYSDAT"
     60  REM RETURN HERE WITH DATE
     70  PRINT D$
     80  PRINT R
     90  END
```

RUN

```
NAM-TSTDAT
10  COM N$(7),R,D$(45)
20  N$="TSTDAT"
30  R=60
40  D$="TWMDYPN"
50  CHAIN "SYSDAT"
60  REM  RET RUNH← HERE WITH DATE
70  PRINT D$
80  PRINT R
90  END

GET-TSTDAT


RUN
TSTDAT

 4:15 PM MONDAY JULY 16, 1973 PORT #11
 730716.

DONE
```

CONTRIBUTED PROGRAM **BASIC**

TITLE: TIME SHARING SYSTEM COMMUNICATION

TALK
36222

DESCRIPTION:

'TALK' enables computer users to report problems, make suggestions, and generally communicate more easily with personnel in charge of a time-sharing system.

Users call the program and enter messages on the teletype under a pre-established ID number. All messages are printed on a file, which is read from time to time by personnel at the computer center. To obtain a printout of the messages, the operator logs in on a user terminal and types RUN-400. This also erases the file so that only new messages will be printed out the next time he dumps the file.

Even though some HP systems have MESSAGE/COMMUNICATE function, there is still an application for this program. The messages will probably averag four or five lines, usually requesting information about a specific progr or problem encountered in programming, or making suggestions as to the operation of the system.

INSTRUCTIONS:

The user need only call the program and enter his message. When he is finished, he types the word END after the question mark, and signs off.

Modify the PRINT statement in line 70 to describe your ID code for user messages.

After the program has been loaded and a file opened, the operator must type: RUN-400. This sets a null character at the beginning of each recor of the file. This need only be done once (immediately after the file is opened), but must be done in order for the program to work.

SPECIAL CONSIDERATIONS:

On a user's system 'TALK' may be broken into two parts. The first part is loaded in the public library; it gives a short description of the program. If the user is interested he calls the program and runs it on any ID-user code. This is done because it is impossible to write onto a file which is in the public library. Then one ID-code is set up and 'TALK' and 'TLKFIL' are placed in the catalog.

This program is useful on a 16-terminal system where the terminals are widely separated. Obviously, for a system where all terminals are located together, this program would serve little purpose.

ACKNOWLEDGEMENTS: John D. Kelley
Babson College

RUN

```
RUN-400
TALK


DONE
RUN
TALK

 'TALK' HAS BEEN MADE AVAILABLE SO THAT ACCOMP USERS CAN REPORT
PROBLEMS, MAKE SUGGESTIONS, AND GENERALLY COMMUNICATE MORE EASILY
WITHIN OUR TIME-SHARING SYSTEM.  THE MESSAGE YOU ENTER IS RECORDED
ON A FILE WHICH IS READ FROM TIME TO TIME BY ACCOMP PERSONNEL.  WE'LL
TAKE WHATEVER ACTION WE CAN TO MAKE ACCOMP COMPUTING MORE EASY FOR YOU.

TO USE 'TALK' PLEASE TYPE:

HELLO-B123,ACCOMP
GET-TALK
RUN


ENTER YOUR MESSAGE AFTER THE QUESTION MARK.  WHEN FINISHED
TYPE 'END' AFTER THE QUESTION MARK.  PLEASE INCLUDE YOUR NAME, ALSO
PHONE NUMBER OR ADDRESS IF YOU EXPECT AN ANSWER.

YOUR MESSAGE:

?THE ACCOUNT NUMBER USED IN THIS SAMPLE RUN IS FICTITIOUS.  YOUR
?T/S CENTER OPERATOR MUST ASSIGN A COMMON ID BY WHICH ALL USERS
?MAY COMMUNICATE WITH HIM VIA 'TALK'.
?END


THANK YOU!

DONE
```

TITLE: SYMBOLIC FILE EDITOR

TIDEX
36204

DESCRIPTION:

TIDEX is used to create, and edit symbolic source files in any HP language. A source file is modified according to commands from a user and stored in a scratch file which may then be copied to the original file. At any given time during the run, not all of the text will be in either the source or scratch file. Some of it will appear on the terminal screen, which is the material that may be modified. A set of commands issued by the user allow modification of the material on the screen, transfer from the source file to the screen, and transfer from the screen to the scratch file. A search operation is also available which involves a transfer from the source to the scratch file while doing the search. This editor was specially written for a CRT and features many cursor control commands.

INSTRUCTIONS:

Declare your source file and scratch file in line 100. (These must have been previously opened.)

Several conventions are used for entering the commands. The input interpreter will check the first character for a period or a semicolon. If a period it will be deleted and the entry assumed to be new text. If a semicolon, it will be replaced by six spaces and the entry assumed to be new source code. Also, if a back-slash appears at the end of a line the entire line will be ignored. Do not use the "ESCAPE" key to delete a line or the program will loose its place on the screen. You may, however, use the shift "0" key for back-space. If the input interpreter cannot see a command in the input it assumes it is to be a new line of text, in which case the line is left on the current cursor position and the cursor moved down to the next non-null line. If the just-used line was the last one on the screen, the screen is rolled up one line, with anything going off the top being written to the scratch file. This allows quick entry of new text. If a command is recognized but the command cannot make sense out of the data supplied to it, the line will simply be erased. Usually, this happens when one is entering new text and the first word happens to be a command. In this case, one must start the line with a period. If one enters a new line of text which is more than 62 characters long, the line will be ignored but left on the screen so it may be easily copied.

Instructions continued on next page.

SPECIAL CONSIDERATIONS:

For a complete listing of the modified text use HPMLUT, 36218.

ACKNOWLEDGEMENTS:

Jonathon Schmidt
Computer Terminal Corporation

INSTRUCTIONS (continued)

In the previous paragraph two new concepts were introduced. One is "cursor position" and the other is "null line". The cursor position will be denoted by where the TSB system prints the question mark indicating a request for more data. At this point, a command or a new line of text may be entered, and the program will take appropriate action. A null line is one which has no characters at all. Note that in this case, a space is considered to be a character, so a line consisting of only spaces is not considered null. Note that since TSB ignores leading spaces, the leading period, (or semicolon) convention mentioned in the above paragraph must be used if leading spaces are desired. When using the leading period, it is advisable to first do a back-space so the line will be correctly justified according to the rest of the screen. The leading period will over-print the question mark. If a leading semicolon is used, the line will be retyped.

In the following list of commands, several other conventions are used. A corner-bracketed item like <M-TEXT> is just a name for what will really be any group of characters in the command. The command consists of exactly the number of characters shown followed by exactly one space. The <SPEC> is assumed to start on the character after the space, so DEL followed by two spaces will try to delete the first line starting with at least one space. This brings us to the concept of a line specification. <SPEC> is a group of characters optionally followed by n periods and m commas. If n and m are not present, the command is performed on the first line with which a match is found. If n periods are present, the first n lines which match are skipped and the command is performed on the next line with which a match is found. If m commas are present, skip m lines after the string match and period skips, and perform the command on the next line. To summarize, if n and m are present, the first n successful matches are thrown away and then m lines are skipped. So "ABC....,," would find the second line after the fourth match on the screen. NOTE: Leading spaces for a match on a screen line are ignored. Most operations are performed on a <SPEC> line, but in the case of the FIND command, the periods and commas are not considered special.

GET-<N> Where <N> is between 1 and 23 get up to n lines from the source file and put them onto the screen. The lines are located starting after CRT the last non-null line on the screen. If the screen is filled before n lines have been obtained, execution of the command is terminated.

ROLL <SPEC> Roll the screen up until the <SPEC> line is at the top. If any lines go off the top of the screen, put them into the scratch file.

SCROLL <SPEC> Identical to ROLL but discard any lines that go off the top of the screen.

MOD <SPEC>↑<M-TEXT>↑<N-TEXT> Replace <M-TEXT>on the <SPEC> line with <N-TEXT>.

MOD <SPEC>↑<M-TEXT>[<N-TEXT> Insert <N-TEXT> after <M-TEXT> on the <SPEC> line.

MOD <SPEC>↑<M-TEXT>]<N-TEXT> Replace all text after <M-TEXT> on the <SPEC> line with <N-TEXT>.

DEL <SPEC> Delete the <SPEC> line.

MOV <SPEC> Move the cursor to the next null line after the <SPEC> line.

COP <SPEC> Copy the <SPEC> line into the current cursor position and move the cursor to the <SPEC> line.

FIND <F-SPEC> Put the current screen contents into the scratch file and erase the screen. Start searching the source file for the F-SPEC line if find, put the found line on the screen. If no find, put the line into the scratch file and get the next line from the source file. If the end of the source file is reached, copy the scratch file into the source, and start searching again, starting at the beginning of the source file. Search until find or until reaching the last line that was on the screen at the beginning of execution of the command in which case print this last line and stop.

FIND BOF Dump screen into scratch and then dump rest of source into scratch and then dump scratch into source. Note, this may be used to copy one file into another.

FIND EOF Dump screen and then rest of source into scratch. Note that this and the above command result in a null screen with the cursor at the top.

END Do a FIND EOF and then terminate run.

END/DEL Just dump the screen into the scratch file and copy the scratch back into the source and then terminate the run.

END/NO Do an END/DEL but do not copy the scratch into the source.

END/DEL/NO Just dump the screen into the scratch file and terminate the run.

LEN Indicate the length of the scratch file so far.

LENGTH Save as above.

PAGE NN ROLL 23 plug GET 23. Allows a user to screen and scan the modified file.

Note that in the case of the LEN or LENGTH command, only exactly those letters specify the command.

```
RUN
OPEN-SCR,90

100 FILES A,SCR
RUN
TIDEX

                        CHARACTERS: 225  LINES: 7    RECORDS: 2
?GET-7
 HARVEY PETE MAUDE JEAN KILEY TRACY LYNN ETHEL DOWNEY STUART CAROL
 PETER SAM
 JOAN PEGGY GEORGE BECKY KATHY
 22-5540
 HP CUPERTINO IN HOUSE TIME SHARING SERVICE
 WELCOMES YOU TO ITS 24 HOUR UP TIME SERVICE AT LOW RATES WITH HIGH
 SERVICE
?MOD WELC↑WITH HIGH]←↑
 WELCOMES YOU TO ITS 24 HOUR UP TIME SERVICE AT LOW RATES

?MOD SERV]←↑SERVICE↑
?END/DEL/NO

                        CHARACTERS: 210  LINES: 6    RECORDS: 2

DONE
R@N←←UN
TIDEX

                        CHARACTERS: 210  LINES: 6    RECORDS: 2
?GET-6
 HARVEY PETE MAUDE JEAN KILEY TRACY LYNN ETHEL DOWNEY STUART CAROL
 PETER SAM
 JOAN PEGGY GEORGE BECKY KATHY
 22-5540
 HP CUPERTINO IN HOUSE TIME SHARING SERVICE
 WELCOMES YOU TO ITS 24 HOUR UP TIME SERVICE AT LOW RATES
?END

                        CHARACTERS: 210  LINES: 6    RECORDS: 2

DONE
```

TIMER
-36297

TITLE: TIME OF THE DAY

DESCRIPTION:

This subroutine returns the time of day in a convenient alpha format.

INSTRUCTIONS:

The time is returned in a string array T$ internally dimensioned for 10 characters.

Output Parameter: T$, the time of day

Entry Point: 9910

Variables Used: T$(10), Z$(10), Z1, Z2, Z3, and Z8

SPECIAL CONSIDERATIONS:

The length of this subroutine is 390 words. This may be reduced to 285 words by deleting the REM statements.

If this subroutine is used in conjunction with DATER, the DIM statement is line 9906 should be deleted. Other than this, the two subroutines are perfectly compatible.

ACKNOWLEDGEMENTS:

Lawrence E. Turner, Jr.
Pacific Union College

```
RUN

LIS
TEST

10  GOSUB 9910
20  PRINT T$
30  STOP

APP-TIMER
9999 END

RUN
TEST

11:31 AM

DONE
```

TITLE
36114

TITLE: CHARACTER GENERATION

DESCRIPTION:

This program generates and positions letters, numbers and some symbols for the Model 7200A Graphic Plotter. There are two programs in this package: TITLE and CHRGEN.

INSTRUCTIONS:

CRE-CHAR,22
RUN CHRGEN to create the file "CHAR".
Get and RUN "TITLE".

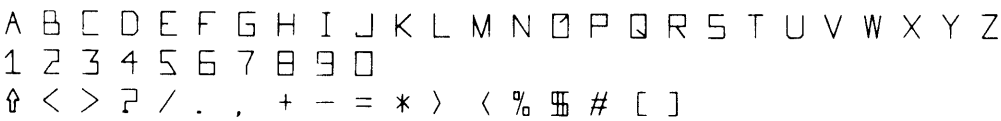
Program responds with:

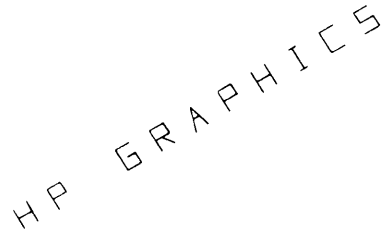
Instructions?	Yes or No (Answer "Yes" the first time you RUN)
Graph Size in Major Divisions:	
Width?	Major Divisions
Height?	Major Divisions
Inputs Desired:	1 or 2 or 3 or 4
Size?	0.1 Major Divisions
Location (X,Y)?	Major Divisions from lower left
Angle?	Degrees from horizontal
TITLE:	
More (1234)?	No or 1 or 2 or 3 or 4

ACKNOWLEDGEMENTS: Hewlett-Packard/San Diego

RUN

```
RUN
TITLE

INSTRUCTIONS?YES

YOU WILL BE ASKED FOR INPUTS:

1=SIZE,2=LOCATION,3=ANGLE,4=TITLE(1234 FOR ALL):

ANY COMBINATION OF THE FOUR NUMBERS MAY BE USED,
SUCH AS '24' FOR LOCATION AND TITLE ONLY.

SIZE IS 0.1 MAJOR DIVISION PER LETTER MULTIPLIED BY
THE NUMBER YOU ENTER; ENTERING '20' WOULD PRODUCE LETTERS
OCCUPYING 2X2 MAJOR DIVISIONS. SIZE IS SET AT '5' IF
YOU DO NOT INPUT.

LOCATION IS THE^POSITION ON THE GRAPH IN MAJOR DIVISIONS,
MEASURED FROM THE LOWER LEFT OF THE GRAPH; 5,5 WOULD
BE THE CENTER OF A 10X10 DIVISION GRAPH.

ANGLE IS THE ANGLE IN DEGREES FROM HORIZONTAL.

TITLE IS THE TEXT YOU WISH PRINTED, UP TO 72 CHARACTERS.
A CONTROL 'O' PRODUCES A CARRIAGE RETURN AND LINEFEED
ON THE GRAPH. A CONTROL 'N' PRODUCES A LINEFEED ONLY.
A CONTROL 'O' AS THE LAST CHARACTER PRODUCES A CARRIAGE
RETURN, LINEFEED, AND A REQUEST FOR MORE INPUT.

GRAPH SIZE IN MAJOR DIVISIONS:
WIDTH ?15
HEIGHT ?10
INPUTS DESIRED:
1=SIZE,2=LOCATION,3=ANGLE,4=TITLE (1234 FOR ALL):?24
LOCATION (X,Y)?0,8
TITLE:
?ABCDEFGHIJKLMNOPQRSTUVWXYZ
?1234567890
?↑<>?/.,+-=*)(%$#[]
PLTL
MORE (1234)?234
LOCATION (X,Y)?8,3
ANGLE?30
TITLE:
?HP GRAPHICS
PLTL
MORE (1234)?NO

DONE
```





TITLE: CREATES FILE "VCHAR"

DESCRIPTION:

This program generates the file NAMed "VCHAR" which contains the characters required for "VSUB", HP 36558.

INSTRUCTIONS:

If the file "VCHAR" is not on your system, OPEN a file NAMed VCHAR to 5 records, and then RUN "UCHARS". The file will be filled with characters necessary to RUN subroutine VSUB.

ACKNOWLEDGEMENTS: Graduate School of Business
Stanford University

```
RUN
CRE
-OPE-VCHAR,5
RUN
UCHARS

@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑← !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[

DONE
```

CONTRIBUTED PROGRAM **BASIC**

VCHART
36555

TITLE: INVESTMENT DECISIONS USING TEKTRONIX 4010

DESCRIPTION:

This program allows the user to make investment decisions over a twenty-five week period, given the previous 25 weeks history of prices for a simulated stock. Alternatively, the values can be considered representative of some overall market index.

INSTRUCTIONS:

See page 2

SPECIAL CONSIDERATIONS:

For detailed instructions for using the Tektronix 4010 display, see "Special Considerations" section of VSUB, HP No. 36558, page 3.

ACKNOWLEDGEMENTS:

Graduate School of Business
Stanford University

INTRODUCTION:

The program must be run using one of the Tektronix graphic terminals. When the program is RUN, the user is asked to select one of several types of stocks (numered 1, 2, ...). Each type corresponds to a model of stock price behavior (e.g., random walk, random walk with upward drift, random walk with downward drift, cyclical curve with random deviations, etc.). The user will, in general, not know while using the program which model corresponds to the type of stock he has selected. The program uses random numbers, so the actual pattern will differ from case to case, even if the same "type" of stock is selected.

After the user selects the type of stock, the program will set up the three charts and plot the **first 25 week's price history, as shown in the example on the following page. The program will then pause for an input.** The cross-hairs should be positioned to indicate the disposition of the user's account between a long and short position. Initially, the account is considered to be "invested" entirely in cash, and to have a value of $10,000. If the investor places the cross-hair at the top of the lower left-hand graph, the money will be invested 100% long. In other words, the entire account will be used to purchase shares of the stock. If he places the cross-hair at the bottom of the graph, the account will be 100% short. In other words, $10,000 worth of the stock will be borrowed and sold at the current price. The cash received will be added to the initial cash balance giving $20,000 in cash, a liability of $10,000 (the current market value of the borrowed stock), and net worth of $10,000. As the price of the stock varies, the value of the liability will also change, as will the net worth.

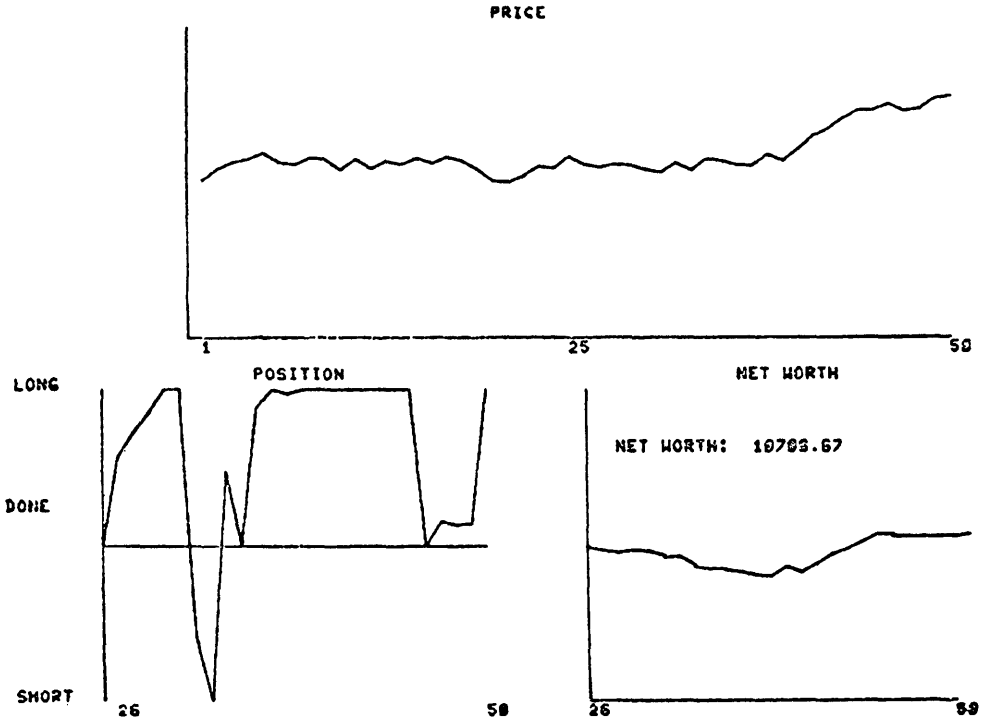
The program credits interest to cash positions at the rate of 6/52% per week. In addition, brokerage fees are charged for any transaction (purchase or sale) at the rate of 1% of the market value.

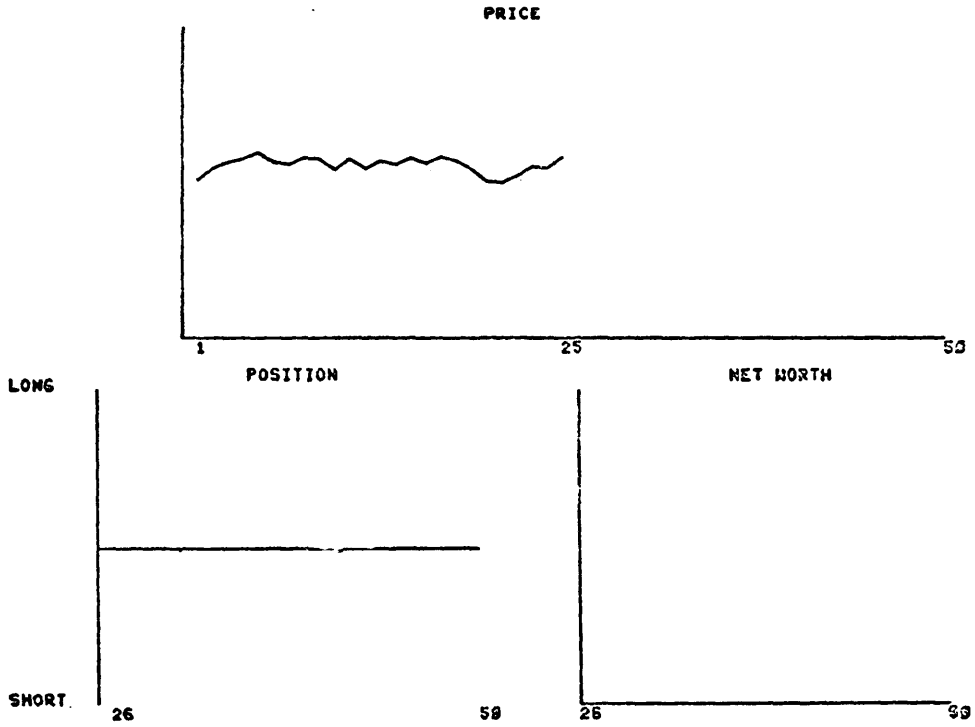
As soon as the user indicates the desired long/short position for the next period, brokerage fees are subtracted from his account, the resultant net worth plotted on the lower left-hand diagram, the next week's price plotted on the upper diagram, and resultant net worth then plotted on the lower right-hand diagram. The user is then asked to select his position for the next week.

When indicating a position, only the vertical location of the cross-hairs is relevant. The selected point will be plotted in the appropriate horizontal location, however, by the program.

When the final week has ended, the program will indicate the net worth of the account. The user may then re-run the program or quit.

RUN




PRICE
1
25
50
POSITION
NET WORTH
LONG
SHORT
26
50
26
50

CONTRIBUTED PROGRAM **BASIC**

VREGPL
36556

TITLE: PLOTTING X AND Y VARIABLES USING TEKTRONIX 4010

DESCRIPTION:

This subroutine allows the user to obtain a plot of the values of X and Y variables in a simple regression, along with the regression line, using the Tektronix 4010 display.

INSTRUCTIONS:

The user must arrange the variables in two lists. Variable N should be set to equal the number of observations. Variables Y(1) through Y(N) should contain the values of the y-variable (vertical axis). Variables X(1) through X(N) should contain the values of the X-variable (horizontal) axis). The subroutine will ask for the values of the remaining variables (e.g., the title, names of the variables, scales for the axes, etc.). The user may override some of the features of the subroutine, if desired, by furnishing values in advance and deleting the appropriate transfer to portions of the code.

The names of the variables should be less than 14 characters long. Values of the variables outside the scale chosen by the user will not be plotted, however, the regression line will be fit to all the data (whether ploteed or not).

If the user indicates that he wants the points connected, the program will draw a straight line from X(1), Y(1) to X(2), Y(2), another straight line from X(2), Y(2) to X(3), Y(3), etc. If a point falls outside the area, it will not be plotted and no line will be drawn to or from it.

Axes are included in the diagram if appropriate (i.e., if the value of zero falls within the selected scale).

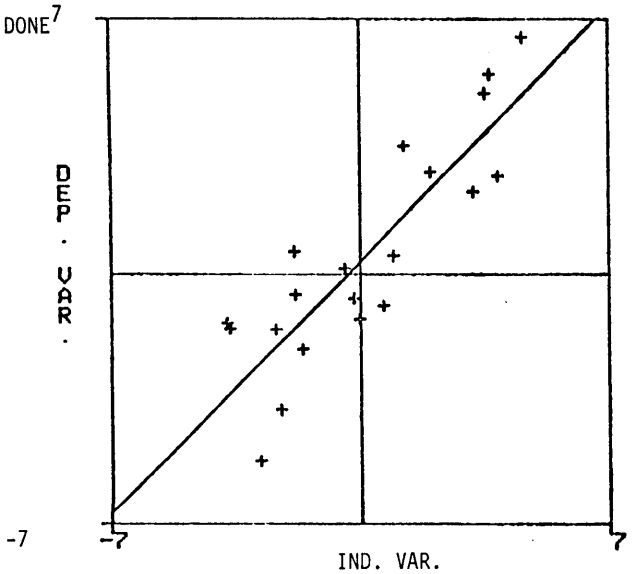
SPECIAL CONSIDERATIONS:

For detailed instructions for using the Tektronix 4010 display, see "Special Considerations", Section of VSUB, HP No. 36558, page 3.

ACKNOWLEDGEMENTS: Graduate School of Business
Stanford University

Note: Because the Tektronix 4010 display was not available, the sample run has been typed.

```
TITLE? SAMPLE OUTPUT

NAME OF X-VARIABLE? IND. VAR.

NAME OF Y-VARIABLE? DEP. VAR.

DO YOU WANT THE REGRESSION LINE PLOTTED? YES
DO YOU WANT THE POINTS CONNECTED? NO

                DEP. VAR.       IND. VAR.
MAXIMUM          6.50415         4.58875
MINIMUM         -5.28261        -3.77222
AVERAGE          .471087         .128865
STD DEV          3.02562         2.60445
(UNADJUSTED)

REGRESSION LINE --
     DEP. VAR.        =     0.34187  +      1.00272*IND. VAR.

STANDARD ERRORS:           0.36055         0.13827
T-VALUES:                  0.94819         7.25207

R-SQUARED -- UNADJUSTED:  .745016        ADJUSTED:  .73085

SCALE FOR IND. VAR.
  LEFT? -7
  RIGHT? 7
SCALE FOR DEP. VAR.
  BOTTOM? -7
  TOP? 7
```



SAMPLE OUTPUT

```
TITLE? TEST OUTPUT

NAME OF X-VARIABLE? TIME

NAME OF Y-VARIABLE? QUANTITY

DO YOU WANT THE REGRESSION LINE PLOTTED? YES
DO YOU WANT THE POINTS CONNECTED? YES
```

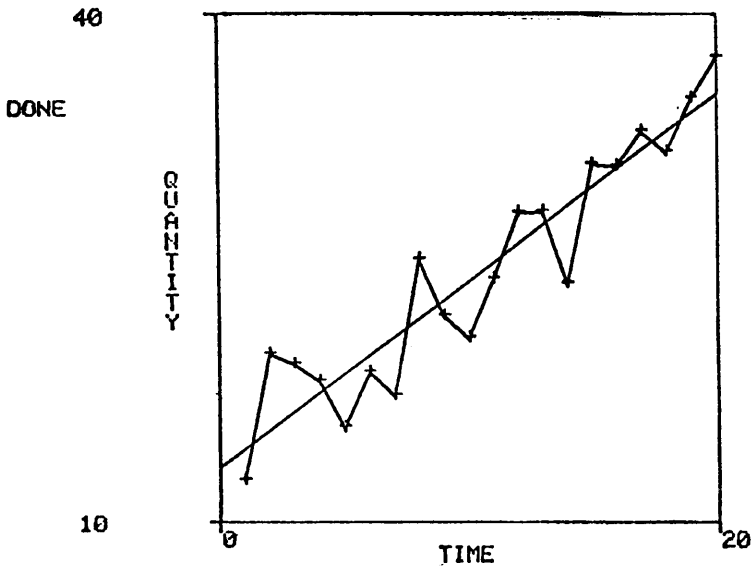
	QUANTITY	TIME
MAXIMUM	37.5622	20
MINIMUM	12.4963	1
AVERAGE	24.8123	10.5
STD DEV (UNADJUSTED)	6.83798	5.76628

REGRESSION LINE --
QUANTITY = 13.15337 + 1.11038*TIME

STANDARD ERRORS:	1.17547	0.09813
T-VALUES:	11.18991	11.31581

R-SQUARED -- UNADJUSTED: .876751 ADJUSTED: .869905

SCALE FOR TIME
LEFT? 0
RIGHT? 20
SCALE FOR QUANTITY
BOTTOM? 10
TOP? 40



TEST OUTPUT

CONTRIBUTED PROGRAM **BASIC**

VSUB 36558

TITLE: DISPLAY ROUTINE USING TEKTRONIX 4010

DESCRIPTION:

This is a set of BASIC subroutines for display operations using the Tektronix 4010 display.

INSTRUCTIONS:

See Page 2

SPECIAL CONSIDERATIONS:

See Page 3

ACKNOWLEDGEMENTS:

Graduate School of Business
Stanford University

INSTRUCTIONS

The routines must read two strings of characters from file $VCHAR. As written, the program reads the information from file #1, which is declared to be $VCHAR in line 5020. If other files are to be used in a program, it is imperative to arrange the FILES statement or statements and/or the file read statement in line 5022 to guarantee that the appropriate information will be read from file $VCHAR.

The subroutines use two string variables -- X$ and Y$ -- for virtually all operations. These variables must not be used elsewhere in the program. Variables X9 and Y9 are used for both graphic input and output. Variables Z8 and Z9 and string variable C$ are also used in several routines.

The subroutines and their functions are described below.

1. Initialize

   GOSUB 5002

   This reads the required characters from file $VCHAR. It must be called before any of the other routines.

2. Erase

   GOSUB 5004

   This erases the screen.

3. Wait

   GOSUB 5006

   This rings the bell, then waits for the user to press the RETURN key. When the key is pressed, the screen is erased and the cursor moved to the upper left-hand corner of the screen.

4. Pen up

   GOSUB 5008

   This places the system in the graphic mode, then moves the cursor to location (X9,Y9). The value of X9 should be between 0 and 1024 (inclusive). The value of Y9 should be between 0 and 780 (inclusive). Values will be rounded to the nearest integer by the routine. The system will be left in graphic mode upon completion.

5. Pen down

   GOSUB 5010

   This moves the cursor to location (X9,Y9), drawing a straight line from the previous location. The value of X9 should be between 0 and 1024 (inclusive). The value of Y9 should be between 0 and 780 (inclusive). Values will be rounded to the nearest integer by the routine. The system should be in graphic mode prior to the use of this routine (i.e., it should be preceded by either a PEN UP or another PEN DOWN operation). The system will be left in graphic mode upon completion.

6. Alpha

   GOSUB 5012

   This places the system in alpha mode. The location of the cursor is unchanged.

7. Graphic input

   GOSUB 5014

   This activates the cross-hairs on the display, then waits until the user presses a key. The location of the cross-hairs at the time is placed in variables X9 and Y9. The character sent by the user will be placed in C$(1,1). The system will be left in alpha mode upon completion.

8. Interpret

GOSUB 5016

This reads information from data statements and produces the indicated output. Before using the routine, the appropriate data statements should be indicated via a RESTORE command. The routine will continue to read data statements until an "E" is reached.

The data statements may contain instructions written in the following rudimentary language:

"D", <x-coordinate> , <y-coordinate>
(go to the specified location with the pen down)

"U", <x-coordinate> , <y-coordinate>
(go to the specified location with the pen up)

"A", <character string>
(shift to alphabetic mode, then print the specified characters)

"E"
(end of instructions)

The following data statements provide the instructions required to draw a box with the word "HELP" inside:

```
100  DATA "U",400,200,"D",400,300,"D",600,300
101  DATA "D",600,200,"D",400,200,"U",480,250
102  DATA "A","HELP","E"
```

SPECIAL CONSIDERATIONS

The following are instructions for using the Tektronix 4010 Display with the Hewlett-Packard 2000C Computer.

Introduction

The Tektronix 4010 Computer Display Terminal allows both graphic and alphanumeric input and output when used with the Hewlett-Packard 2000C computer system. The display can be connected directly to the computer or used via telephone lines with acoustic couplers. The Stanford Graduate School of Business operates four displays; each is connected directly to the computer system and located in room B05 of the GSB building. Transmission to the computer system is generally performed at a speed limited by the user's typing ability. Transmission from the computer system can be accomplished at speeds as great as 240 characters per second.

Operating the Display

The power switch is located on the stand beneath the keyboard (approximately at the point where the user's knee touches the stand). Allow a few seconds after turning it on for the system to warm up.

The switch marked "LOCAL/LINE" at the top of the keyboard should be in the "LINE" position.

The key marked "PAGE" erases the screen and returns the cursor (blinking square) to the upper left-hand portion of the screen. No information is sent to the computer. After the system has warmed up, PAGE should be pressed to clear the screen.

After clearing the screen, log in to the computer system in the normal manner. All keys are operated as on a teletype with a few exceptions. The key marked ALT MODE is used to indicate an error in an entire line (it serves the same function as the key marked ESC on most teletypes). The dash obtained by pressing the SHIFT key and the (letter) O key indicates the deletion of the preceding character (it serves the same function as the left arrow (←) on most teletypes. The BREAK key may be used to interrupt a running program or a listing.

The cursor indicates the position at which the next material will be written. Pressing the RETURN key will move the cursor to the left and down one line. Pressing the RESET will move the cursor to the upper left-hand corner of the screen, without sending any information to the computer.

SPECIAL CONSIDERATIONS (continued)

Information may be written on the screen using the standard set of uppercase letters, digits, and special characters. A total of 35 lines of 72 characters each may be displayed at one time.

The system may also be used to plot lines and points by placing the system in the graphic mode. Locations are referenced via a coordinate system, with (0.0) indicating the lower left-hand corner. Up to 1024 points may be differentiated in the horizontal direction, and up to 780 points in the vertical direction.

The system may also be used to indicate the location of a point on the screen. The computer program activates a set of cross-hairs which appear on the screen. The user may move these to any point on the screen with the thumbwheels located on the right-hand side of the keyboard. When the intersection is at the desired point, the user simply presses a key (e.g., "X"). The coordinates of the cross-hairs at the time are then sent to the computer.

When a permanent copy of a display is desired, the user should press the key marked MAKE COPY. This will activate the hard-copy unit connected to the four terminals, producing an 8" x 10" version of the material on the screen.

Use with Standard Programs

The display may be used with standard programs as a teletype replacement. However, unless some provision is made, the display will soon become full and information will be lost and/or written over other information. The screen may be cleared by pressing the PAGE key when needed. However, to use the display more effectively, delays and/or automatic erase commands should be inserted in the program. The VSUB routine may be used for this purpose.

Use with Display-Oriented Programs

A number of programs have been written expressly for use with the display. Each begins with the letter V and is included in the HP BASIC Handbook (e.g., VCHART).

Displacement of Characters

If the graphic routines are used to position the cursor prior to "printing" a character, the center of the character will be located above and to the right of the referenced location. To center a character on a given location, some displacement is required. For example, to center a decimal point at location X9,Y9, set:

```
X9 = X9 - 4
Y9 = Y9 - 2
```

Required displacements for 4 characters commonly used for plotting follow:

Character	X-Displacement	Y-Displacement
X	4	7
.	4	2
*	4	6
+	4	6

CONTRIBUTED PROGRAM **BASIC**

TITLE: TIC-TAC-TOE ON THE TEKTRONIX 4010 DISPLAY TERMINAL

VTTT
36559

DESCRIPTION:

This program allows the user to play a game of 4-by-4-by-4 tic-tac-toe against the computer.

INSTRUCTIONS:

When run, the program draws the "boards" on the Tektronix 4010 display, then waits for the user to make his move. The user indicates the desired square by positioning the cross-hairs appropriately, then pressing the "X" key. The user's move will be marked with an X. The computer will then move, indicating its square with an "O". The user may then make his next move, etc.

To win, you must have four squares in a straight line. All four may be on the same "board", or they may be on different boards. Diagonal lines are eligible, as are horizontal and vertical.

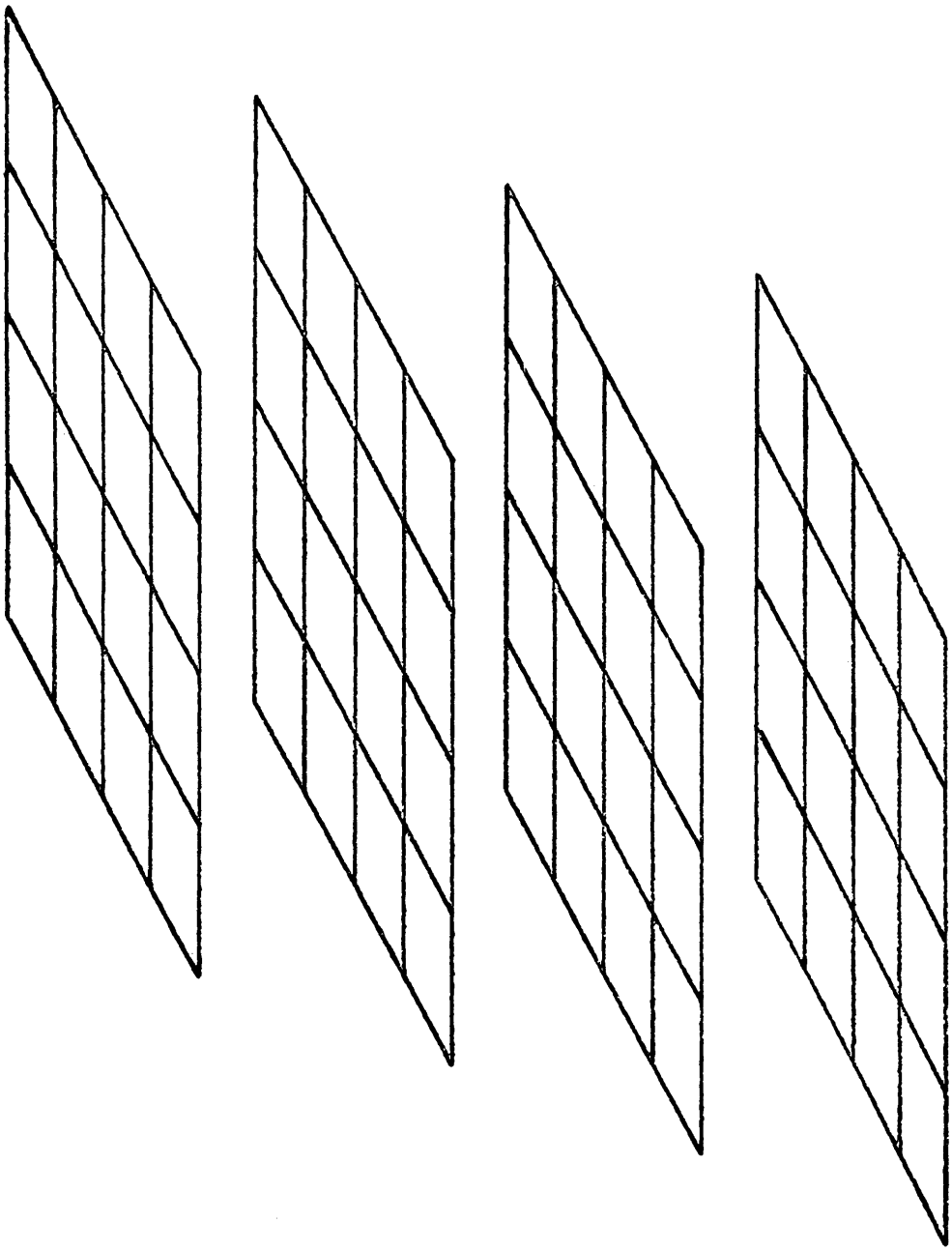
When either the user or the computer wins, the winning squares are connected with a straight line and the program terminates.

The "boards" are shown on the following page.

SPECIAL CONSIDERATIONS:

For detailed instructions for using the Tektronix 4010 display, see "Special Considerations", Section of VSUB, HP No. 36558, page 3.

ACKNOWLEDGEMENTS: Graduate School of Business
Stanford University



CONTRIBUTED PROGRAM **BASIC**

TITLE: MANUAL/TAPE FILE LOADER AND DUMP PROGRAMS

XTRACT
36221

DESCRIPTION:

XTRACT and TAPDUM are programs which perform file input and output from/to paper tape.

INSTRUCTIONS:

XTRACT

1. All characters representing a standard numeric format are interpreted as numbers. In addition fractional exponents will be properly interpreted.

2. Strings are defined as all nonnumeric formats (and numeric formats when the first character of the input line is a backslash--see 3.).

3. The blank is the unit separator. Single and multiple blanks are ignored (except as separators) unless the first character of an input line is the backslash, in which case the entire line is interpreted as a single string. The backslash conveys no editing information unless it occupies the first position in an input line (otherwise it is interpreted as any other character).

4. Input is terminated in the standard fashion with a control-C.

5. Pause time for the ENTER statement (160) is arbitrarily set at 100 seconds (15); it may be appropiate to reset this time limit for recycling at the maximum of 256 seconds to allow for slow typers.

TAPDUM

1. Use of this program provides a compact punched random file dump up to the first logical or physical end-of-file mark in a form that is compitable for file loading with XTRACT.

SPECIAL CONSIDERATIONS:

XTRACT is designed to accept keyboard input as indicated above, or to read paper tapes that are prepared by the program TAPDUM.

ACKNOWLEDGEMENTS: William W. Moss
Charlottesville, VA

```
RUN


100 FILES A
RUN
XTRACT

THIS IS A TEST OF THIS FILE LOADING SYSTEM
\THIS LINE IS READ AS A SINGLE STRING
THIS ONE IS NOT
NOR IS THIS ONE
THESE     SPACES     ARE     IGNORED
\   THESE       ARE       NOT
NUMBERS:  1  2  3  -45  1E12  -1.5E-31  -37.789E-26.365  1E1.678
\END OF TEST


DONE



GET-TAPDUM
100 FILES A
RUN
TAPDUM

YOU HAVE 10 SECONDS TO TURN ON THE PUNCH


THIS IS A TEST OF THIS FILE LOADING SYSTEM
\THIS LINE IS READ AS A SINGLE STRING
THIS ONE IS NOT NOR IS THIS ONE THESE SPACES ARE IGNORED
\   THESE       ARE       NOT
NUMBERS:  1     2     3     -45     1.00000E+12    -1.50000E-31
-1.63069E-25     47.6431
\END OF TEST


DONE



GET-FILIST
100 FILES A
RUN
FILIST

IS T/S AN HP 2000'A', 'B', 'C', 'E', OR 'F'?C

STOP LISTING FILE 1    AT THE FIRST EOF (Y OR N OR Q)?Y


FILE 1    RECORD 1

THIS           IS             A              TEST           OF
THIS           FILE           LOADING        SYSTEM
THIS LINE IS READ AS A SINGLE STRING         THIS           ONE
IS             NOT            NOR            IS             THIS
ONE            THESE          SPACES         ARE            IGNORED
   THESE       ARE       NOT  NUMBERS:        1     2     3     -45
 1.00000E+12   -1.50000E-31   -1.63069E-25     47.6431     END OF TEST
END OF FILE 1

STOP LISTING FILE 2    AT THE FIRST EOF (Y OR N OR Q)?Q

DONE
```

TITLE: DUMP FILE TO DATA STATEMENTS — DATA 36287

DESCRIPTION:

This program dumps the contents of a file into BASIC "DATA" statements. It inputs a starting and step value for the statement numbers. The program "READ" is included as a sample program to illustrate filling a file ("TEST") with data.

INSTRUCTIONS:

The file "CHARS" (HP 36220) must be in the system library, containing at least the first 58 ASCII characters in order in one string. The user's file may be filled using "READ", "FILOAD" (HP 36010) or a similar format. The DATA program prompts the user to enter the file name, and the starting statement number (SS) and interval (I). The paper tape punch is then turned on, and the paper tape is generated. The tape dump should include leader and no "X OFF" characters on the tape.

SPECIAL CONSIDERATIONS:

The file CHARS can be filled by ASSIGN statement masks.

ACKNOWLEDGEMENTS:

Bruce A. Robinson
The Evergreen State College

READ

```
10  FILES TEST
20  DIM A$[72]
30  GOTO TYP(0) OF 40,70,150
40  READ A
50  PRINT #1;A
60  GOTO 30
70  READ A$
80  PRINT #1;A$
90  GOTO 30
100   DATA 123.124,383,0.59,-293.45,-3.8E-12,"NOW IS THE "
110   DATA "FOR ALL GOOD MEN TO COME TO THE AID OF THEIR PARTY"
120   DATA 3.73774E+21,3.5E+16,-3.7E-15,1.E-13
130   DATA "THIS STRING IS 72 CHARACTERS LONG 567890123456789012345678901234567890123456789012"
140   DATA 0,0,0,1,2,3,4,-1,-2,-3,-4,"OVER AND OUT"
150   END
```

RUN

```
RUN
FILE?TEST
SS,I?500,1
TEST

500   DATA   123.124,383,0,59,-293.45,-3.80000E-12,"NOW IS THE "
501   DATA   "FOR ALL GOOD MEN TO COME TO THE AID OF THEIR PARTY"
502   DATA   +3.73774E+21,+3.50000E+16,-3.70000E-15,+1.00000E-13
503   DATA   "THIS STRING IS 72 CHARACTERS LONG 567890123456789012345678901234567890123456789012"
504   DATA   0,0,0,1,2,3,4,-1,-2,-3,-4,"OVER AND OUT"
```

CONTRIBUTED PROGRAM **BASIC**

TITLE:	BASIC LANGUAGE PROGRAM CROSS-REFERENCE GENERATOR	XREF 36143
DESCRIPTION:	XREF accepts as input an XPUnched tape of a BASIC program, and outputs a cross-reference of line numbers and a dictionary of variables with line references. GOSUB references are preceded by an S. Lines longer than 72 characters are not processed, but are separately listed on the bottom.	
INSTRUCTIONS:	"XPUNCH" the program needing a cross-reference index. Place the paper tape in the reader and RUN the program XREF.	
SPECIAL CONSIDERATIONS:	Terminal must have an auto tape reader.	
ACKNOWLEDGEMENTS:		

```
RUN

RUN
XREF

PROGRAM NAME?STGINT
DO YOU WANT A CROSS REFERENCE OF LINE NUMBERS?YES
DO YOU WANT A DICTIONARY OF VARIABLES WITH LINE REFERENCES?YES
DO YOU WANT THE LISTING DOUBLE SPACED?NO
PUT THE XPUNCHED TAPE IN THE READER AND SET IT TO AUTO.
9000  REM  ***  HP TIME-SHARED BASIC PROGRAM LIBRARY  ******************
9001  REM  ***
9002  REM  ***     (A104)  36176  REV A 1/72
9003  REM  ***     STGINT: SUBROUTINES TO CONVERT BETWEEN STRINGS AND
9004  REM  ***           POSITIVE INTEGERS
9005  REM  ***
9006  REM  ***  CONTRIBUTED PROGRAM  **********************************
9020  REM
9030  REM  **  ENTER SUBROUTINE WITH Z9= (MAX) LENGTH OF STRING
9040  REM      AND Z$ DIMENSIONED IN MAIN PROGRAM.
9050  REM   *  IF A STRING TO NUMBER CONVERSION, ENTER WITH Z8=0 AND
9060  REM      'Z$' AS THE STRING.  THE NUMBER IS RETURNED AS 'Z'
9070  REM   *  IF A NUMBER TO STRING CONVERSION, ENTER WITH Z8=1 AND
9080  REM      AND 'Z' AS THE NUMBER.  THE STRING IS RETURNED AS 'Z$'
9090  REM
9100  DIM Y$[10]
9110  Y$="0123456789"
9120  IF Z8=0 THEN 9310
9130  IF Z8=1 THEN 9160
9140  PRINT "Z8 NOT PROPERLY SPECIFIED"
9150  STOP
9160  IF Z<0 THEN 9290
9170  IF INT(Z)#Z THEN 9290
9180  IF Z>((10↑Z9)-1) THEN 9270
9190  Z2=Z3=Z4=0
9200  FOR Z1=Z9 TO 1 STEP -1
9210  Z3=Z3+1
9220  Z2=INT(Z/(10↑(Z1-1)))-10*Z4
9230  Z4=(10*Z4)+Z2
9240  Z$[Z3,Z3]=Y$[Z2+1,Z2+1]
9250  NEXT Z1
9260  RETURN
9270  PRINT "NUMBER TOO LARGE FOR STRING LENGTH"
9280  STOP
9290  PRINT "NEGATIVE OR NON-INTEGER NUMBER DETECTED"
9300  STOP
9310  Z=0
9320  Z7=Z9
9330  IF Z$[Z7,Z7]#" " THEN 9360
9340  Z7=Z7-1
9350  GOTO 9330
9360  Z2=0
9370  FOR Z1=Z7 TO 1 STEP -1
9380  IF Z$[Z1,Z1]>"4" THEN 9430
9390  FOR Z3=0 TO 4
9400  IF Z$[Z1,Z1]=Y$[Z3+1,Z3+1] THEN 9480
9410  NEXT Z3
9420  GOTO 9460
9430  FOR Z3=5 TO 9
9440  IF Z$[Z1,Z1]=Y$[Z3+1,Z3+1] THEN 9480
9450  NEXT Z3
9460  PRINT "NON-NUMERIC CHARACTER DETECTED."
9470  STOP
9480  Z=Z+Z3*10↑Z2
9490  Z2=Z2+1
9500  NEXT Z1
9510  RETURN
9520  END

IS THERE ANOTHER XPUNCHED TAPE FOR STGINT?NO


PROGRAM NAME IS:    STGINT


     CROSS REFERENCE OF LINE NUMBERS (S = GOSUB REFERENCE):
```

```
LINE NO.     REFERENCED BY:
  9160     9130
  9270     9180
  9290     9160  9170
  9310     9120
  9330     9350
  9360     9330
  9430     9380
  9460     9420
  9480     9400  9440

 DICTIONARY OF SIMPLE VARIABLES WITH LINE REFERENCES:

   Z       9160  9170  9180  9220  9310  9480
   Z1      9200  9220  9250  9370  9380  9400  9440  9500
   Z2      9190  9220  9230  9240  9360  9480  9490
   Z3      9190  9210  9240  9390  9400  9410  9430  9440  9450  9480
   Z4      9190  9220  9230
   Z7      9320  9330  9340  9370
   Z8      9120  9130
   Z9      9180  9200  9320

 DICTIONARY OF STRING VARIABLES WITH LINE REFERENCES:

   Y$      9100  9110  9240  9400  9440
   Z$      9240  9330  9380  9400  9440

 LIST OF LINES WITH MORE THAN 72 CHARACTERS
 (AND HENCE UNPROCESSED):

  9006


DONE
```