

JANUARY 1978

 Data General

**PROGRAM
LIBRARY
CATALOG**

USER'S  GROUP



TABLE OF CONTENTS

		Page No.
Introduction		v
	Program No.	Program Name
Section 1	COMMUNICATIONS	
	56	RTOS 2702 Emulator (IBM) 1-1
	93	Auto Caller Driver Software 1-1
	95	UT200 Simulator 1-1
Section 2	COMPUTATIONAL	
	1	Fast Fourier Transform and Inverse 2-1
	5	Eigenvalue Solution Program 2-1
	13	BCD to Binary Conversion by Radix Reduction 2-1
	22	Focal Length by Microscope Method 2-1
	25	Conversion of Linear English Measurement Values to Metric Equivalents 2-1
	26	Conversion of Metric Linear Measurement to English Equivalents 2-1
	40	Polynomial Curve Fit (Polyfit) 2-2
	44	CURVEFIT 2-2
	45	ARCTANGENT 2-2
	50	RTLAB (Reaction Time Lab Program) 2-2
	52	WHOLE 2-2
	53	Single Precision and Double Precision BCD to Binary 2-2
	75	Acid-Base Titration 2-2
	81	BASIC Curve Fit 2-2
	84	Gasoline Economy Program 2-3
	92	Foreign Exchange 2-3
	111	Calculator 2-3
	126	Analysis of Variance Package 2-3
	128	Stanford Business School Package 2-4
	134	Correlation Program 2-5
	148	ALGOL Date Routines 2-5
	156	POLAR 2-5
	162	Function Diagram Printing Program 2-5
	164	ASCII Fast Integer Tangent 2-5
	184	POLYFIT 2-5
	185	EXPFIT 2-6
	205	RRAND 2-6
	222	ALFFT. LB 2-6
	223	ALPLOT. LB 2-6
Section 3	EDUCATION	
	46	DACBAP (Dana College Basic Algorithm Processor) 3-1
	76	Class Grading Program 3-1
	77	Lab Grades 3-1
	120	Co-Pilot 3-1
Section 4	GAMES	
	27	BASIC Football and Baseball Games 4-1
	58	BATNUM 4-1
	59	SLOT 4-1
	62	BLACKJACK 4-1
	64	TURTLERACE 4-1
	65	LUNAR 4-1
	66	HORSERACE 4-1
	67	HEMAN 4-1
	72	2-Player Football 4-1

	78	FORTTRAN Qubic	4-2
	79	BASIC Qubic	4-2
	86	BATTLESHIPS	4-2
	87	STAR TREK	4-2
	96	NIM	4-2
	124	BIORHYTHM	4-3
	129	NOVA Radio Music Interpreter	4-3
	158	DOTS	4-3
	188	C/13 (a Chess Game)	4-3
	216	ALGOL Biorhythm Chart	4-3
Section 5		INSTRUMENTATION AND CONTROL	
	15	Pin List System	5-1
	19	Wire List Layout Program	5-1
	24	Conductor Selection Program for Low Voltage Three Phase Systems	5-1
	37	TAD (Tape to/from analog conversion program)	5-1
	113	D-A Converter Plotting Subroutines	5-2
	125	SOS Extended BASIC with RTC and Digital I/O	5-2
Section 6		UTILITIES	
	3	Teletype Tape Edit	6-1
	4	BEAP Editor/Assembler Program	6-1
	10	Punch Card to Tape Converter	6-1
	11	NOVA Magnetic Tape Input/Output System	6-1
	12	Hollerith to ASCII Conversion Subroutine	6-1
	17	Input/Output Package	6-2
	20	Overlay Editor - Assembler Program	6-2
	21	RESUB	6-2
	23	Single User BASIC with Matrices and Strings	6-2
	28	Machine Language I/O Subroutines for T. S. BASIC	6-3
	31	NOVA Operating System Program	6-3
	33	EMUL-8	6-3
	35	Core Compare and Load	6-3
	36	Binary Tape Mapper	6-3
	38	Binary Tape Merger	6-3
	41	Combo Binary Relocatable Editor	6-3
	42	Fast Special Format Floating Point Subroutines	6-4
	43	Incremental Plotter Subroutines for BASIC	6-4
	48	Memory Test Program	6-4
	49	Assembler Source Program Analyzer	6-4
	55	Floating Point Sort	6-4
	68	Error Reporter	6-5
	69	Disk Copy (Rev .01)	6-5
	71	NOVA Accounting System, Main Log Routine	6-5
	80	PRINTX	6-5
	82	General Purpose FORTRAN Macrogenerator	6-5
	85	Cassette/Mag Tape Reversible Read Subroutines	6-6
	88	RTOS Snapshot Dump	6-6
	89	TASK-S Simple Multi-Programming Supervisor	6-6
	90	BASIC Printing Utilities	6-6
	91	Strip (String Processor)	6-7
	97	MUMPS	6-7
	100	RDOS/FORTRAN IV I/O Subroutines	6-7
	101	Character Processing Routines	6-7
	102	RTOS/RDOS System Utilization Task	6-7
	103	DTECT	6-7
	104	Calendar	6-8
	105	F1130-IBM 1130 FORTRAN Program Converter	6-8
	106	Disassembler	6-8
	107	MCA Binary Loader	6-9
	108	Filecompare, Binary - Pack	6-9

109	NOVA Interpreter	6-9
110	FORTRAN Random Number Generator	6-9
115	BASIC Editor, BASIC Single Line	6-10
117	RDOS Binary Loader	6-10
119	PREP	6-10
123	FORTRAN Post Processor	6-10
131	Foreground CLI	6-10
133	Symbolic Alphanumeric Generator	6-11
136	MYPRI	6-11
138	EBCDIC to ASCII Translator	6-11
139	PRNT - Listing Routine	6-11
140	ALGOL Edit and Dump Package	6-11
141	BASIC Line Editor	6-11
145	FORTRAN IV Get Channel Routine	6-11
146	UNICO	6-11
147	IBMCO	6-11
149	CLEAN	6-12
150	MAPCK	6-12
151	EIAAS	6-12
152	FORTRAN Error Message Interceptor	6-12
153	Directory Listing Program	6-12
154	L. M. U. Package	6-13
155	BRTIOS (A-D I/O and Logical) Functions for the BASIC User)	6-16
157	PAGE	6-16
159	Program Space Memory Sizer	6-16
161	CRT to Disc Data Formatter	6-16
163	FORTRAN IV COM. CM Read Routine	6-16
165	EX UTILITY	6-16
199	CHAT - a File Attribute Changing Program	6-17
200	COMPILE - (Multiple FORTRAN Source Programs)	6-17
201	DISKBOOT	6-17
202	SBRTB	6-17
203	T COPY (Tape Copy Program)	6-17
204	MTUTIL (DG Mag Tape Operation Programs)	6-17
206	GANDER (AOS KEY TO DISK, DATA ENTRY PROGRAM)	6-17
209	ASM16 (an IMP-16 Cross Assembler)	6-17
210	Source Level Symbolic Debugger for ECLIPSE FORTRAN 5	6-18
211	Driver for LC/50 Card Reader/Punch/Printer	6-18
214	LISP 1.5 Interpreter	6-18
215	BASIC Accounting File Summary	6-18
217	ALEX. AL-An Automatic ALGOL Execute Program	6-18
218	ALGLIB-An Extended LFE Program	6-18
219	ALGPROC-A Collection of ALGOL Procedures	6-19
220	SCRIBE-An ALGOL Text Format Program	6-19
221	TANKA-A Demonstration of String Manipulation Techniques	6-19
Appendix A	Program Library Price List	A-1
Appendix B	Program Request Form	B-1
Appendix C	Installation Membership Form	C-1



PROGRAM LIBRARY CATALOG

INTRODUCTION

The purpose of the Data General User Group is to promote the effective use of Data General computer systems through a program of information and software exchange. This catalog contains an abstract of each user-submitted program currently available to members of the User Group. A small fee is charged for each program to help defray reproduction, distribution, and administration costs.

Data General warrants only that the programs supplied are duplicates of the programs submitted by the program author. Therefore, Data General will not be liable for any special, indirect, or consequential damages arising out of, or in connection with, the use of the materials.

The Program Library functions only as a clearinghouse and distributor for User-contributed Programs, and as such, no technical or programming assistance is available from Data General. If a program does not work as described in the abstract, the problem should be fully documented and sent to the User's Group Administrator, Data General Corp., Westboro, MA 01581. Such critiques will be forwarded to the program author for comment. If a program is not usable in the form supplied and described in the documentation, it will be removed from the Library.

The description and availability of the programs contained in this catalog are subject to change without notice. Distribution of programs will be in accordance with the current policy of the Data General User Group.

Users are urged to submit programs to the Library which they feel would be of use to other members. In return for each program submitted (and accepted), a User may select a program from the catalog at no charge.

PROGRAM DELINEATIONS

In order to make the identification of a particular type of program convenient for the User, the following classification system has been adopted.

COMMUNICATIONS - Programs used to transmit or control data flow from the originating point via common carrier or private communications lines to a remote location for processing or distribution. Also includes communications between processors.

COMPUTATIONAL - Programs used in computing for applications such as scientific, engineering, bio-medical, military, aerospace, and all other non-manufacturing areas.

EDUCATIONAL - Programs that are specifically useful to educational institutions. Generally these programs relate to the instructional process, administrative procedures and courseware. Programs that educators or students also will find useful will be listed in other categories, especially Computational, Games, and Utilities.

INSTRUMENTATION AND CONTROL - Programs which will be utilized directly in on-line control of continuous or batch manufacturing processes, or associated with the support of general manufacturing activity, and other non-scientific data acquisition and control systems.

UTILITIES - Programs which facilitate the generation and operation of other programs, e.g., interpreters, editors, assemblers, simulators, I/O handlers. Also the facilitation of the movement of data, for example: tape to disk, disk to tape, disk to disk, etc.

ORDERING PROCEDURE

Use the order blanks supplied in this catalog and forward with a purchase order specifying the catalog number and the name of each program requested. Please do not send cash. Mail Orders to:

Data General Corporation
Data General User Group
Route 9 MS82541
Westboro, Massachusetts 01581

WHAT YOU GET

Whenever possible, source files are supplied. In some cases, only binary files are available. (This fact is noted in the program abstracts.) Most programs will be accompanied by a User-Supplied Documentation File. Some of the BASIC Programs are self documenting. Also, unless otherwise noted, memory requirements are specified in **K Words**.

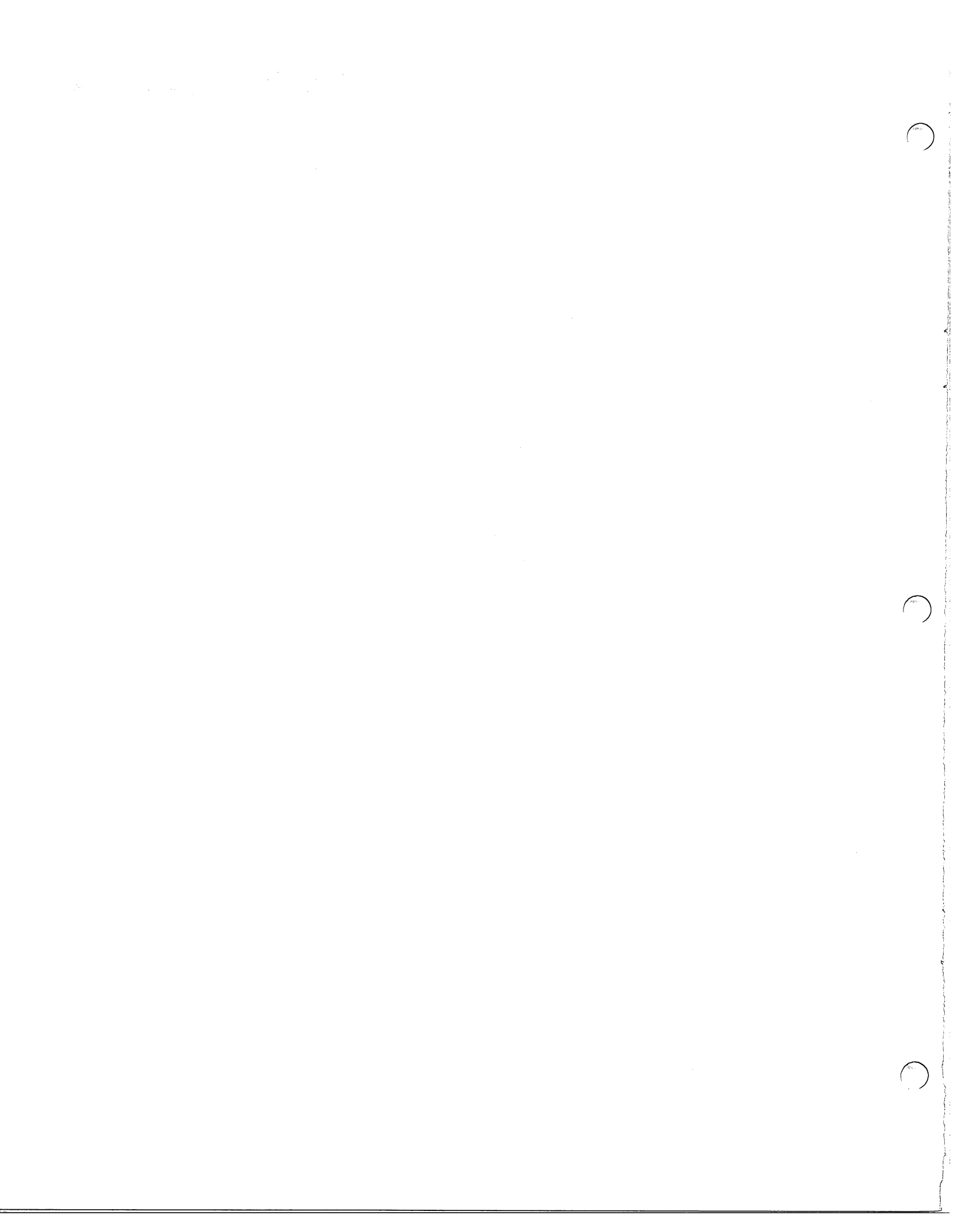
Listings or Documentation Files are not separately available.

MEDIA

Programs are available on paper tape and 800 or 1600 bpi, 9-track magnetic tape. If no media is specified for an order, programs will be supplied on paper tape.

* Please note that a minimum order of \$50.00 is required.

COMMUNICATIONS



Catalog No. 220-056

RTOS 2702 EMULATOR (.IBM)

.IBM is a special purpose software driver for the DGC 4025 360/370 Interface unit designed to be used with the DGC Real Time Operating System. It is designed to emulate the functional characteristics of an IBM 2702 with a Telegraph Type II adapter. This includes recognizing and responding to the channel command set for the 2702, providing ending status to commands and terminating READ operations on receipt of certain control characters. Data is transferred to and from the host in the byte multiplex mode. Calling sequences for .IBM are similar to other RTOS device handlers and are outlined in the .IBM user's manual.

4K read/write memory and RTOS are required. Necessary peripheral is the 4025 360/370 interface. The program is written in ASSEMBLY language.

Catalog No. 220-093

AUTO CALLER DRIVER SOFTWARE

This program provides an interface to a Bell auto call unit using a Model 5280 controller from Data General's Custom Products group.

- 1) The program will run on **Rev. 3** RDOS or later. Otherwise, modify .UIEX for user interrupt return.
- 2) If device codes are different, change ACU = Definition on page 1 and reassemble.

Catalog No. 220-095

UT200 SIMULATOR

This program allows a NOVA family computer to communicate with a remote CDC 6600/6400 computer to simulate a 200-user terminal. Jobs may be submitted from the card reader in the normal manner, or from any input device supported by the operating system, including disk files. An assembly time option allows for generation of a communications history file.

All of the routines in the package must be assembled with the Macro Assembler and loaded with the relocatable Math Library (099-000001-02). The programs require 16K. They were developed and tested using RDOS **Rev. 3**. The host operating system was SCOPE EXPORT/IMPORT. A modified CDROV.RB is necessary and included in the package.

The routines operate at 2400 baud; any higher rate is untested.



COMPUTATIONAL



Catalog No. 220-001

FAST FOURIER TRANSFORM AND INVERSE

This potentially reentrant program is in subroutine form and may be called from a main program. The JSR statement is followed by the following parameters: forward (F(N*T))/ inverse (F(L*T)) transform switch (0 or 1); number of sample points; the real parts of the input series. When control returns from the subroutine, the transformed data will have replaced the input data.

The number of sample points must be an integer power of 2, up to 1024_{10} complex values. Trigonometric computations are performed by table look-up, and input data is scaled, to reduce the possibility of overflow, with the scaling factor retained to allow for restoration of the output series.

Storage: 610_{10} words (6 on page 0, 348 for routine, 256 for cosine table).

Catalog No. 220-005

EIGENVALUE SOLUTION PROGRAM

Using the Jacobi method of Eigenvalue solution, this BASIC program, written to run on Multi-User Basic (DGC no. 091-000026) solves for the eigenvalues and eigenvectors of any NxN matrix. The utilization of the matrix manipulation capability in this 71-statement program enables data to be supplied in one DATA statement:

E. G. 540 DATA N,A(1,1), A(2,1), . . . , A(N,N).

A printout of the A matrix's eigenvalues and associated eigenvectors is produced in a form suitable for such scientific applications as chemistry, physics, and the mathematical sciences.

Catalog No. 220-013

BCD TO BINARY CONVERSION BY RADIX REDUCTION

This ASSEMBLY language routine reduces the 4-digit BCD value in AC0 to an equivalent binary representation, replacing the original BCD value in AC0. The packed BCD string is treated as a binary value with an "expanded" radix. The routine reduces the radix of each digit until the converted value is obtained. This technique is considerably faster than the isolation of each digit followed by the binary reconstruction.

Catalog No. 220-022

FOCAL LENGTH BY MICROSCOPE METHOD

Using approximate thick lens equations for symmetric lenses, this BASIC program calculates the focal length of symmetric lenses from four microscope measurements from a micrometer indicating microscope. Printout includes the average and standard deviation of the centered focal lengths of a batch of lenses.

Catalog No. 220-025

CONVERSION OF LINEAR ENGLISH MEASUREMENT VALUES TO METRIC EQUIVALENTS

This BASIC language program converts English Linear Measurement values into their Metric System equivalents. Since the computer will not accept any fractional values except those expressed in decimal form, all input data must be expressed as whole feet and/or inches, and decimal fractions thereof.

Four input options are accepted. Data may be entered in feet only (to the nearest whole feet or decimal fraction thereof); feet and inches only (to the nearest whole inch or decimal fraction thereof); inches only, (to the nearest whole inch or decimal fraction thereof); or in miles and/or fractions, thereof.

Output may be obtained in Millimeters, Centimeters, Meters, or Kilometers, by entering the proper value of B when starting a run.

Another feature of this program gives it capability to convert long strings to values and identify the results by groups.

Catalog No. 220-026

CONVERSION OF METRIC LINEAR MEASUREMENT TO ENGLISH EQUIVALENTS

This BASIC program converts Metric linear measurements, millimeters, centimeters, meters, or kilometers into the English linear measurement equivalent of inches or feet. Provision is made for 17 different conversion options. The user may have his output in inches, to the nearest inch. In this case, the computer will round off all fractional values; those less than 1/2 are truncated; those greater than 1/2 are raised to the next whole number. Decimal values may be obtained, or common fractions to the nearest 1/2, 1/4, 1/8, 1/16, 1/32, or 1/64. If the output is desired in feet, feet and inches, or feet, inches, and fractions thereof (decimal or common) similar options may be had.

Catalog No. 220-040

POLYNOMIAL CURVE FIT (POLYFIT)

POLYFIT will fit least-square polynomials to bivariate data using an orthogonal polynomial method. Limits are 11th-degree fit and a maximum of 100 data points. The program allows user to specify the lowest degree polynomial to be fit, and then fits the polynomials in order of ascending degree. At each stage, the index of determination (r^2) is printed, and the user has the choice of going to the next higher degree fit, seeing either of two summaries of fit at that stage, or of stopping the program.

POLYFIT is written in extended BASIC and requires only a teletype for operation.

Catalog No. 220-044

CURVEFIT

CURVEFIT will determine which of 6 curves best fits the User's data using a least squares linearized fit. The program accepts up to 200 observations on variables.

The program is written in Extended BASIC and requires only a teletype for operation.

Catalog No. 220-045

ARCTANGENT

This program will take the fixed point single precision ARCTANGENT of the $\sin(t)/\cos(t)$ and return the result in AC3 in the range $-\pi \leq \phi \leq \pi$. ARCTANGENT is called as a subroutine with the $\sin(t)$ in AC0 and $\cos(t)$ in AC1. Return to instruction following call with AC3 = ARCTANGENT.

Either Hardware Multiply/Divide or Software Multiply/Divide is necessary. ARCTANGENT is written in ASSEMBLY language.

Catalog No. 220-050

RTLAB

RTLAB is a package of 20 relocatable ASSEMBLY language routines for on-line experimentation in discrete-trial behavioral experiments where classification of response or reaction time are dependent variables. The routines of RTLAB perform such functions as entering and storing parameters, preparing an appropriately counterbalanced sequence of trials for each subject, controlling the presentation and timing of the events on each trial, scoring and storing subject's responses, providing summary statistics for a subject when the session is over, and transferring data to an appropriate storage medium. The package is constructed to allow maximum User flexibility, especially in mating RTLAB with the User's experimental task and specific instrumentation. RTLAB is not designed for time-sharing between several subject stations.

8K of memory is necessary. Only peripheral required is a teletype, others are up to the User.

Catalog No. 220-052

WHOLE

This assembler subroutine, callable from FORTRAN IV, will truncate the fractional part of a double precision real number and return with only the non fractional part. The user must supply his own mainline program.

Catalog No. 220-053

SINGLE PRECISION AND DOUBLE PRECISION BCD TO BINARY

This ASSEMBLY language subroutine converts either an 8-digit BCD-value in AC0, AC1 (double precision, entry: BCDBD (or a 4-digit BCD-value in AC0 (single precision, entry: BCDDBS (into its equivalent binary representation replacing the BCD-value. The radix reduction algorithm (User's Group catalog no. 220-013) is used to convert a single precision 4-digit value. In double precision conversion, the high order part is converted first, then multiplied by 10000 (decimal) and the converted low order part is added to obtain the final result. The routine is not re-entrant. This 36 (decimal) word routine may be loaded anywhere in core.

8K of memory is needed.

Catalog No. 220-075

ACID-BASE TITRATION

This BASIC program handles up to 100 points of acid-base titration data. Provision is made for correction of input data before the start of processing. End point volume and pH is returned. By user choice, the program can be applied to standardization of base, determination of acid concentration, or equivalent weight of the acid. Options include a complete table of normalized data including first derivative -- with end point indicated -- as well as a teletype plot of the titration curve. The program will run with Data General BASIC in 8K of core.

Catalog No. 220-081

BASIC CURVE FIT

This program performs a least squares curve fit on the data provided it (by keyboard or test equation). The number of points handled is dependent on core available. Polynomials fitting is possible to degree 10. After fit, the operator has the option to obtain a printer plot, statistics, or a new fit of the data.

Catalog No. 220-084

GASOLINE ECONOMY PROGRAM

This program accepts data input from the console regarding fuel usage in an automobile. Mileages may be input either from a standard cumulative odometer, or from a "trip" odometer, which is reset to zero each time fuel is added to the tank. A list of average mpg at each fill-up is printed, along with a value for the cruising range, and if desired, the user may then delete any suspicious data points. Then an analysis of fuel economy by brand is printed, and the user is permitted to compute economies over portions of his data as needed. Cost can, of course, be substituted for fuel used, and in that case a cost analysis by brand will be obtained.

The program is coded in FORTRAN. Minimum configuration is a NOVA-line CPU with 12K or more memory and teletype.

Catalog No. 220-092

FOREIGN EXCHANGE

This BASIC program will compute the Foreign currency equivalence of one U.S. dollar. It includes most of the popular European and Far Eastern currencies. Exchange rate information needed for input can be obtained through any bank or large newspaper.

Catalog No. 220-111

CALCULATOR

This program simulates a four function (+, -, /, *) calculator having one storage register. The expression is typed on the teletype and when a carriage return is hit, the expression is evaluated from left to right. The expression value is then either typed out or stored in the storage register. The storage register may be used in place of a number in an expression. (Its latest value is substituted when the expression is evaluated.)

Output may be either in octal or decimal (decimal is the default), and is under the control of the user. Input numbers can be specified either in octal or decimal, with decimal being the default.

All calculations are in integer, having a range of -32,768 to +32,767.

The program is written in ASSEMBLY language and may be run with any DG operating system using the hardware multiply/divide board if available. It is about 550 words in length and requires no page 0 locations.

Catalog No. 220-126

ANALYSIS OF VARIANCE PACKAGE

This package of BASIC programs performs eight analysis of variance routines (ANOVA) including:

- 1) Independent one-way ANOVA
- 2) Two way ANOVA
- 3) Three way ANOVA
- 4) Randomized blocks ANOVA
- 5) Two factor randomized blocks ANOVA
- 6) Split-plot factorial ANOVA (1 between, 1 within)
- 7) ANOVA for a split-plot factorial (2 between, 1 within)
- 8) Split-plot factorial ANOVA (1 between, 2 within)

STANFORD BUSINESS SCHOOL PACKAGE

This package consists of 36 programs written in (rev. 3) Extended BASIC. Although originally written for MBA students at Stanford, the routines should be useful in a variety of situations. RDOS with 32K and Extended BASIC is required. Below are brief descriptions of each program.

NAME	DESCRIPTION
GWBULL	Subjective probability-random values
GCPM1	Critical path analysis
GLPSA1	Linear programming two-phase simplex method
GSPMG	Stanford portfolio management game
GSSS	Small system simulator
VCHART	Investment decisions
GK COST	Price/earnings ratio calculations
GKASSF	Warrant price calculation
G N M R V B / GMRGB	Securities portfolio analysis and determination
GVOTE	Committee choice analysis
GTASPD	Selective probability distribution
GDPA	Efficient "corner" portfolios
GDAPI	Abnormal performance index
VRRC	Investment strategy analysis
GRGPLT	Simple regression and plot
GCPATH	Critical path analysis
GVPDQT	Plotting data

GTHOR	Securities EPS growth
GINTLP	Linear programming - variable restricted to one or zero
GSIMEQ	Simultaneous linear equations
GMCR01/ GMCR02	Fiscal policy game
GMCR05/ GMCR06	Economic policy game
GRANK	Ranking statistics
GANOVA	Analysis of variance
GLP	Linear programming
GNETFL	Network flow determination (Max/Min)
GFNRAT	Financial ratios
GRISKA	Risk analysis in capital investment
GSTKVL	Stock valuation
GIRRPV	Investment return (cash flow)
GCHLIN	Rating investment funds
B2AGPDQ	Utility program
B2AGSP5	Utility program

A manual is available as documentation that contains a detailed description of each program, the listing, and a sample run.

Catalog No. 220-134

CORRELATION PROGRAM

This program calculates and prints a correlation matrix and a t-score matrix for as many variables as the output device allows (11 for a TTY). Features include: data entry via the TTY or the PTR, optional listing of the input data, optional subroutine overlay that allows repeated transformations of the input data (log, SQR, 1/x, arcsine, and multiplication by a constant), and the listing of the variable means at the end of the analysis. The number of observations per variable is a factor of the amount of core the user has available. The program itself requires 2504 bytes of storage.

Catalog No. 220-148

ALGOL DATE ROUTINES

These ALGOL procedures handle calendar dates, and can be used to calculate time intervals over extended periods.

- STRING PROCEDURE NMDATE returns a string with calendar information, e.g., Mon. 1 Jan. 1900, from a given Gregorian date.
- INTEGER PROCEDURE JULDAY returns the modified JULIAN date number from a given Gregorian date.
- PROCEDURE GREGDAY returns three integers for day, month, and year (Gregorian Calendar) for a given Julian Date Number.
- PROGRAM TSTJDAY .AL tests the above procedures, and gives time intervals in days for consecutive keyboard inputs.

Catalog No. 220-156

POLAR

POLAR is a basic program designed to plot polar functions on a CRT, teletype, or line printer. The function to be graphed is converted to a basic statement that is entered within a loop. There are several options for this program, including the size of the graph, the domain of the angle, and the density of the points. The asterisk (*) is the character used to represent a point on the graph.

Catalog No. 220-162

FUNCTION DIAGRAM PRINTING PROGRAM

This BASIC language program prints up to 300 points of 1 to 10 different functions represented by characters chosen by the user. Points are defined by array /X,Y/ and Y. Data may be input in any order by the dialog procedure, or by procedures provided by the user. The program performs sorting and prepares data for line-printer output, shows the user extreme figures of X and Y coordinates and proposes the size of the drawing and scale of printing steps.

Catalog No. 220-164

ASCII FAST INTEGER TANGENT

This subroutine written in ASSEMBLY language calculates the tangent $m = 10^4 \tan(N/10^4)$ of an integer value $N/10^4 \leq 10^4$ in less than 55 MICROSEC. on a NOVA 1200 computer with hardware multiply-divide. The accuracy is better than ± 1 of the least significant digit. The routine needs 40 memory locations.

Catalog No. 220-184

POLYFIT

POLYFIT is a utility program which fits a polynomial of degree K ($\leq KMAX$) to a set of data $\{x_i, f_i\}$ $i=1, \dots, N$ by the method of least squares. It is based on the linear least squares algorithm using orthogonal polynomials. Each data point is weighted inversely to the square of the standard error, Δf_i , of the dependent variable.

POLYFIT is written in ALGOL (REV. 2.03) and will run on any NOVA or ECLIPSE processor with RDOS (REV. 5.00).

Catalog No. 220-185

EXPFIT

EXPFIT is a utility program which fits the exponential function $f(x) = A + B \exp(-Cx)$ to a set of data $\{x_i, f_i\}$, $i=1, \dots, N$ by the method of least squares. It is based on a special purpose non-linear least squares algorithm. Each data point is weighted inversely to the square of the standard error, δF_i , of the dependent variable.

The program is written in ALGOL (REV. 2.03).

Catalog No. 220-205

RRAND

RRAND generates floating-point numbers of greater quality than those available from the FORTRAN V random-number routines which are limited by their integer seed and has poor resolution and repetition after the 2^{16} number has been generated. RRAND uses a large table of 32-bit seeds which provides for better resolution and less number repetition. In addition, RRAND saves its "state" information on disc; future runs may restore this state and pick up where the sequence left off, or they may start from scratch for predictable operation during debugging.

Minimum configuration: Any RDOS system 747₈ or 787₁₀, and the standard FORTRAN V library.

Catalog No. 220-222

ALFFT.LB

ALFFT.LB is a library of ALGOL procedures implementing the Discrete Fourier Transform and selected data transformations based on that transform. The implementation is based on the Fast Fourier Transform Algorithm for complex and real data sequences.

Written in ALGOL, the program requires 24K of memory, a NOVA computer, any disk and any console, and operates under RDOS Rev. 5.00 and up.

Catalog No. 220-223

ALPLOT.LB

This is a library of ALGOL procedures for plotting graphs of computed data on Tektronix 4000 Series visual display units. ALPLOT.LB is based on the FORTRAN package GPLOT.LB. Users familiar with that package will find that ALPLOT uses identical procedure calls. ALGOL users, however, need not be familiar with GPLOT.LB or FORTRAN since this program is written and documented as a self-contained and independent library package.

ALPLOT.LB runs under RDOS REV. 5.00 and up. Minimum hardware required is any NOVA processor, with any disk and Tektronix 4000 Series Visual Display Units.

EDUCATION



Catalog No. 220-046**DACBAP (Dana College Basic Algorithm Processor)**

This program simulates a single accumulator integer computer. It is especially useful for teaching beginning programmers some of the typical operations of a digital computer in a simplified, pseudo-ASSEMBLY language. Following training in BASIC, it functions as a transition to a lower-level language. The program should be self-explanatory to a teacher who knows something about ASSEMBLY language coding, e.g. use of instructions like load, store, etc.

12K of memory is necessary. The program was written to run on Time Share BASIC, but the author thinks it will run on Single User BASIC as well.

Catalog No. 220-076**CLASS GRADING PROGRAM**

This BASIC program will process class grades for up to 140 students. Input data includes student number, three exam scores and a final exam score. The weight of the final exam is determined by the user. A negative entry for any exam is not counted. Student averages are determined first by averaging the three exam scores and then weighting with the final exam. Provision is made for correction of entries before processing. The corrections list is terminated by a negative entry number. Two output tables are provided: 1) All grades and the average are listed in "gradebook" order, i.e., the same order as input. 2) Grades are listed in order of student number with each student assigned a rank in class. (This table is suitable for posting). On both tables mean and standard deviation are also indicated for each exam and the average. To facilitate processing intermediate grades, a data tape may be punched. Subsequent scores may then be entered as corrections. The program will run on any Data General BASIC in 8K core.

Catalog No. 220-077**LAB GRADES**

This BASIC program will process laboratory grades for up to nine experiments; laboratory examinations are optional. Individual experiments may be weighted; examination weight is determined by the user. Input data includes student number, letter grades for up to nine experiments and exam average, both numerical score and letter grade. Negative entries are not counted. Provision is made for correction of entries before processing. The corrections list is terminated by a negative entry number. Letter grades are input using the following code: F:0, D:1, C-:2, C:3, C+:4, B-:5, B:6, B+:7, A-:8, A:9, A+:10. The output table includes student number, exam average (numerical score and letter grade), experiment average (letter grade), and course letter grade. Multiple copies may be output. Grades awarded: N (incomplete), D, C, C+, B, B+, A, A+. This program will run on any Data General BASIC in 8K of core.

Catalog No. 220-120**CO-PILOT**

CO-PILOT is a computer language designed for use by teachers writing computer assisted instruction (CAI) lessons. Some of its features include: selective branching to route students through different materials according to their responses, storage of student responses for later insertion in the lesson, and search for a selected word in a student response. No previous programming experience is required to develop and write the programs.

There are two separate routines: THE CO-PILOT program itself, which runs the lesson, and the CO-PILOT Editor, in which the lessons are written, edited, and revised. A manual is also available that explains the CO-PILOT commands, the editing commands, and the run-time procedure.



GAMES



Catalog No. 220-027

BASIC FOOTBALL AND BASEBALL GAMES

This set of BASIC language programs may be run on Data General's Single User or Time Shared BASIC Interpreters. The set of three tapes includes one Baseball Game and one Football Game (as played by American rules), and a Canadian Football Game.

Catalog No. 220-058

BATNUM

This BASIC language program may be run on Data General's Single User or Multi User BASIC Interpreters. BATNUM is a number elimination game; see how many tries it takes you to guess the correct number.

Catalog No. 220-059

SLOT

This BASIC language program may be run on Data General's Single User or Multi User Interpreters. SLOT is a game that simulates the operation of a slot machine.

Catalog No. 220-062

BLACKJACK

This BASIC language program may be run on Data General's Single User or Multi User BASIC Interpreters. This program simulates a Blackjack game, allowing players to bet.

Catalog No. 220-064

TURTLERACE

This BASIC language program may be run on Data General's Single User or Multi User BASIC Interpreters. Pick your favorite turtle, make a bet, then watch them run (?). TURTLERACE was written for an Infoton CRT and may need modifications to run on another CRT.

Catalog No. 220-065

LUNAR

This BASIC language program may be run on Data General's Single User or Multi User BASIC Interpreters. LUNAR is a simulation of a spaceship landing on the moon. You are the astronaut and must guide your craft to a safe landing.

Catalog No. 220-066

HORSERACE

This BASIC language program may be run on Data General's Single User or Multi User BASIC Interpreters. The program simulates a horserace. Make your bet, and then they're off! All your favorites: Man O'War, Seabiscuit, Citation, are there. HORSERACE was written for an Infoton CRT and may need modifications to run on another CRT.

Catalog No. 220-067

HEMAN

This BASIC language program may be run on Data General's Single User or Multi User BASIC Interpreters. Can you make the bell ring and win a cigar? For this game it is the RETURN key, not a sledge hammer, that measures your strength. HEMAN was written for an Infoton CRT and may need modifications to run on another CRT.

Catalog No. 220-072

2-PLAYER FOOTBALL

The game is played by Canadian rules in that there are only 3 downs, the field is 110 yards, and a touchback yields a single point. Operation of the game is self explanatory.

Hardware Requirements:

- 1) 2 Teletypes \$TTO and \$TTO1
- 2) 24K Memory
- 3) Real Time Clock
- 4) Moving Head Disk

Software Requirements:

- 1) RDOS (3181)
- 2) FORTRAN IV compiler and runtime libraries (3212, 3213)

Catalog No. 220-078

FORTRAN QUBIC

FORTRAN Qubic is a programmed three-dimensional game of TIC-TAC-TOE. The computer plays against the person at the console. The routines may be arranged to provide medium to very difficult competition.

Qubic consists of the main program (FQUB), the I/O routine (FQUF), and four playing routines. Selection of one of the four combinations produces a game of a certain difficulty. Below they are listed in order of increasing difficulty:

- PLAYA, PLAYB1 (Moves by weighting factor without lookahead)
- PLAYA, PLAYB
- PLAYA1, PLAYB
- PLAYA1, PLAYB (Uses tree search by lookahead techniques for sequence of moves that can't lose)

The tree search is to a depth of 10 to a maximum of 1500 nodes expanded.

The game requires 16K and RDOS or RTOS. It is written in FORTRAN IV.

Catalog No. 220-079

BASIC QUBIC

QUBIC is a three-dimensional game of TIC-TAC-TOE programmed in BASIC. The computer plays against the person at the console. The program does not look-ahead but does select moves based on weighting all currently occupied winning lines. The game presents a medium level of difficulty.

The program requires Extended BASIC for operation.

Catalog No. 220-086

BATTLESHIPS

This BASIC language program may be run on Data General's Single User BASIC Interpreter. BattleShips is a naval battle between 2 players. Each player has five different types of ships to choose from and a handicap system is provided for beginners. The ocean is a 60 x 50 rectangle with an island in the Northwest corner. Both players place their ships anywhere they wish. The plays are made up of one ship moving (throttle and direction) and the other ship taking 2 shots involving: gun direction, elevation and number of charges. A map is printed out after each set showing ship positions and shell landing. (The North-South position is given by a line number in the left column). The amount of fuel and number of charges per ship are limited. One direct hit or three near hits will sink the enemy.

Strategy and skill become determining factors with a little experience. Capsizing, colliding, grounding, and falling off the edge of the earth are all possibilities.

Catalog No. 220-087

STAR TREK

Star Trek is an advanced battle-simulation game based on the T. V. series of the same name.

At the computer console, you are captain of a starship of the Federation of Planets. Your ship is of the Enterprise-class; the largest, and most powerful ships in the galaxy. In the simulation, you will battle Klingons, Romulans, and a host of other "enemy" starships.

You have at your command phasers, photon torpedoes, matter-antimatter probes, navigational computers, auto-track radar, and an enormous reserve of power to operate these and other functions. All together, you can give any of 29 different commands.

There are more possibilities in the end of the game than mere win or lose. A total victory is where you have either completely destroyed the enemy, or better yet, forced him to surrender without killing his whole crew. (Romulans have never been known to surrender). A tactical victory is one where you have out-run, out-maneuvered, or exhausted the fuel supply of the enemy. A moral (phyrric) victory is where you self-destruct rather than surrender; or, in blowing up, manage to take the enemy with you. A loss is when you are either destroyed, or surrender. (Romulans have never taken prisoners).

This sophisticated Extended BASIC simulation requires in excess of 20K bytes of user space and a CRT.

Catalog No. 220-096

NIM

This BASIC program may be run on any Data General computer with the Extended BASIC Interpreter. NIM is a game in which two players pick up sticks from piles on the ground. See if you can beat the computer by being the one to pick up the last stick in the game. A clever and persistent player should beat the computer fairly regularly.

Catalog No. 220-124

BIORHYTHM

This program is based on the theory that human lives move in predictable undulations involving three separate cycles:

- A) Physical 23 Days
- B) Emotional 28 Days
- C) Intellectual 33 Days

There are three major areas within the chart to be observed:

- A) Minus
- B) Zero
- C) Plus

All you do is enter your name, date of birth, and the number of months you want plotted and out comes your biorhythm chart.

The program is written in FORTRAN IV and BASIC.

CATALOG No. 220-129

NOVA RADIO MUSIC INTERPRETER

This program plays music on a radio held near a NOVA computer by generating radio frequency interference with the memory switching currents. A demo tape is included with the interpreter and five songs:

- 1) Gentle on my Mind
- 2) Impossible Dream
- 3) Windy
- 4) Walking in the Sunshine
- 5) Penny Lane

An assembly parameter tape is also provided for the simple transliteration of other musical scores.

Catalog No. 220-158

DOTS

A game of DOTS written in BASIC and designed to be played on a teletype or an ADOS terminal. The program needs at least 16K of available core to run. In the game of DOTS you compete with the computer or another person for the most boxes.

Catalog No. 220-188

C/13 A CHESS GAME

C13 is a program that plays chess. It allows all moves including en passant and underpromotion.

C13 is available as an absolute binary to be loaded by the binary loader or RDOS/DOS BOOT.SV. It requires a 4KW NOVA with console on device code IB/11.

C13 is available as a .RB file to be RLDRed under DOS or RDOS. It requires a 4KW memory partition and a console.

C13 is not available as a source.

Catalog No. 220-216

ALGOL BIORHYTHM CHART

This biorhythm charting program is similar to Catalog No. 220-124, but is written in ALGOL.

Single precision arithmetic is used in this program to compute the number of days lived by the subject. A base date of 1/1/1870 allows the program to be accurate until the latter part of August, 2049.



INSTRUMENTATION AND CONTROL



Catalog No. 220-015

PIN LIST SYSTEM

The Pin List System is designed to allow documentation and implementation of logic systems which use wirewrap interconnections. The name and location of each pin is entered on line into the computer. A paper tape is then made of this source list. The source list is then re-entered to produce a sorted listing and a sorted paper tape, both sorted by name and location. The name-sort is used for wirewrapping, and the location-sort is used for error checking and debugging. A system including 4K Nova-line computer is capable of accepting up to 400 individual points, with expansion being a function of core. The two supplied programs generate the pin list, sort, and punch tapes sorted by name and location respectively.

Catalog No. 220-019

WIRE LIST LAYOUT PROGRAM

The wire list generation system is designed to allow the user to quickly layout the wiring list for a group of printed circuit cards that have wirewrap interconnections. The existing system allows the wiring list for eight 86 pin cards to be generated. Each pin is described by six ASCII characters. The program has nine commands:

- BLANKS - Blank the storage area.
- ENTER - Enter data through the teletype.
- READ - Read data from the high-speed reader.
- CHANGE - Change data from the teletype.
- DELETE - Delete a card entry from the table.
- LIST - List all card data on the teletype.
- LIST 1 - List one card data on the teletype.
- PUNCH - Punch data on the high-speed punch.
- WIRE - Layout an optimized wiring diagram.

This program operates in 4096 words of core memory using all locations up to 7630(8).

Catalog No. 220-024

CONDUCTOR SELECTION PROGRAM FOR LOW VOLTAGE THREE PHASE SYSTEMS

This BASIC language program selects either copper conductor or copper feeder duct busway to satisfy the data supplied. Data consists of the following:

- L: Three phase volt-amp load
- V: Line to line voltage
- F: Load power factor in decimal form
- B: Circuit length in feet
- M: Maximum tolerable voltage drop in percent

Two steps are involved in selecting the proper conductor. The first involves a calculation of the load current and an initial selection of a conductor to satisfy this ampacity. Secondly, the conductor is checked to satisfy the maximum tolerable voltage drop. If the voltage drop is excessive, the program finds the first larger size that satisfies the drop requirement. This conductor (or busway) is the desired result of the program.

Catalog No. 220-037

TAD (Tape to/from analog conversion program)

TAD is an analog (A/D converter) to-tape and tape-to-analog (D/A converter) program. The program accepts a 12-bit word from an A/D converter and reformats the data to be stored on a 7-track magnetic tape deck. The program also reads the stored data on tape and outputs it in the correct format to a 12-bit D/A converter. The interrupt system is used to allow the transfers. An external clock is supplied to the A/D converter to provide the start-to-convert pulse.

TAD is an ASSEMBLY language program requiring 4K of memory. Necessary peripherals include: A/D, D/A, 7 track magnetic tape, teletype, and an external clock for A/D converter.

Catalog No. 220-113

D-A CONVERTER PLOTTING SUBROUTINES

These subroutines are designed to provide the user with a flexible plotting capability using a two-channel digital to analog converter system and a two-dimensional voltage recording device, such as an oscilloscope or X-Y recorder. Routines are provided for the drawing of a labelled axis, output of character strings and real numbers, scaling sets of numbers, output of single points, and output of lines connecting points. Communication between the computer and the external interface is left to the user to program, thus making the subroutines quite general in nature. As supplied, these routines will operate any recording device which has a recording mode which does not need repeated renewal, i.e., these routines will operate a storage oscilloscope or a conventional pen control circuit, but will not operate an X-Y recorder equipped with an event marker, or an oscilloscope without the storage option.

Routines are provided to be used with programs written in DGC FORTRAN IV or DGC Extended BASIC. The FORTRAN routines are contained in a library tape set, and the BASIC routines are supplied as source code to be edited into the user's program or ENTERed along with the user's program. The axis-drawing subroutine is not implemented in BASIC.

Catalog No. 220-125

SOS EXTENDED BASIC WITH RTC AND DIGITAL I/O

This is an assembler subroutine to give CALL statements in SOS (Rev. 03) Single User Extended BASIC for real time functions of clocking and digital I/O. Four interrupt driven software clocks provide timing to an accuracy of 10 milliseconds. Each clock can be independently zeroed or have its contents read into the user program. Digital I/O is handled on a pair of Type 4066 boards with access at the whole word or individual bit level. In addition, the console data switches and the output registers can be used for sensing input. The program is provided as a source tape that is assembled and loaded to produce a new working version of SOS BASIC. It is a convenient vehicle for teaching process control or running a variety of experiments.

UTILITIES



Catalog No. 220-003**TELETYPE TAPE EDIT**

This is a paper tape editing program which can co-exist in a 4K system with the Paper Tape Assembler (DGC 091-000002) by residing in the symbol table space. Without affecting the assembler, a paper tape may be edited on the teletype by loading the original tape in the teletype reader, starting the tape edit program, and turning on the teletype punch. With data switch 15 up, a single character is copied only when any one of the other 15 data switches is inverted. Thus, toggling a data switch copies at a controlled rate to permit omission or insertion of characters.

Storage: 14₁₀ words

Catalog No. 220-004**BEAP EDITOR/ASSEMBLER PROGRAM**

BEAP is a combination editor and assembler program with primary applications on systems having only a teletype for I/O. BEAP's 10 keyboard commands enable the programmer to enter his source program into a storage buffer from either the teletype keyboard or reader, examine, modify, output, and assemble the source program in the buffer any number of times without reloading BEAP or the source program.

Programs may be written and debugged in locations 0 through 77 and 7100 thru 7600 without affecting BEAP, the storage buffer, or the binary and bootstrap loaders. BEAP requires approximately 1500₁₀ words of storage and has a buffer capacity of 3700 characters.

Catalog No. 220-010**PUNCH CARD TO TAPE CONVERTER**

This program converts cards punched in IBM 029 format to ASCII formatted punched paper tape. If data switch 0 is up, output is directed to the high speed punch. Absolute locations 15134, 15135, and 15136 may be initialized to contain the following information, respectively: number of columns of the card to be converted; number of columns at the beginning of the card to be ignored; number of lines per page---between Form Feed punches. Version supplied resides in the upper 4K module of an 8K system configuration (S. A. -- 15000).

Catalog No. 220-011**NOVA MAGNETIC TAPE INPUT/OUTPUT SYSTEM**

This system program enables core images to be stored on, and retrieved from, magnetic tape by simple teletype commands. Communication with the system is by file-name (operator generated). Tape searches are done for the filename, and files retrieved are loaded into core with termination options of HALT or automatic program start (analogous to start - block features of the DGC Binary Loader). The system is also stored on magnetic tape and is "booted" into core by a special bootstrap program which replaces the DGC Binary Loader in core and has a S. A. of X7777. Minimum system configuration is 8K of memory, teletype, and one 9-track magnetic tape unit.

Catalog No. 220-012**HOLLERITH TO ASCII CONVERSION SUBROUTINE**

This ASSEMBLY language subroutine, with a calling sequence of JSR HOLL, converts a 12-bit Hollerith code in AC0 to its equivalent 8-bit, even parity ASCII code. The ASCII code is returned in AC0 and all other accumulators are modified by the subroutine. The Hollerith code is assumed to be a card image, right justified, with the "12" punch in bit position 4. The 64₁₀ work subroutine is self-contained and may be assembled anywhere in core.

Catalog No. 220-017

INPUT/OUTPUT PACKAGE

This is a package of ten subroutines to facilitate I/O operations with the teletype. Subroutines are provided for:

- 1) Character input from TTY
- 2) Character output to TTY
- 3) Generation of N spaces
- 4) Generation of CR and LF
- 5) Output of character string into storage
- 6) Reads a character string from TTY into storage
- 7) Reading of a signed, single or double precision, octal or decimal numeral string from TTY and conversion into binary.
- 8) Branching on reception of a break character
- 9) Conversion of a binary number into signed or unsigned, decimal or octal, numeral string and printing on TTY
- 10) Formatting of numbers generated by routine no. 9. These subroutines reside in 742₈ memory locations.

Catalog No. 220-020

(LP)

OVERLAY EDITOR-ASSEMBLER SYSTEM

An executive routine, a modified Data General editor, and a modified Data General absolute assembler make up this overlay system. The editor stores a text buffer in core and the assembler will assemble the contents of this text buffer. Routines have been added to the editor which facilitate source compression (by deleting comments) if core storage is not adequate. In between overlay operations, other programs may be run in low core (at least up to address 4463) and the text buffer may still be recovered. Core above address 4463 is divided into the assembler user symbol table and the text buffer. The system may be run with a teletype and 4K of memory, but a high speed paper tape reader and at least 8K of memory allow convenient operation.

Catalog No. 220-021

(LP)

RESUB

RESUB, Relocatable Extended Single User Basic is a BASIC Interpreter System with features to allow concurrent operation with background programs. RESUB is a modified version of Data General's Time Share Basic. The following features should be useful in several single user applications:

- 1) Tape, Card, Dump, and Punch commands. These allow loading and dumping programs with user supplied routines.
- 2) Subroutine Call - RESUB'S SUBROUTINE call may be used as a console command. It is also possible to pass a string variable to a subroutine.
- 3) Stack processor for handling nested interrupts.
- 4) Provisions for handling background programs when BASIC is not running or at bottom of interrupt stack.
- 5) Subroutines may be initialized at Start Up.
- 6) Location 20 contains address of first unused location in upper core.
- 7) Entry points are supplied for setting Run Time accumulator, substituting line printer for teletype, etc.
- 8) No location below 20 is used until RESUB is started. Consequently, it should be possible to load under DOS.

RESUB is about 14,000₈ words long. About 20 words in page zero are available.

The minimum configuration required for using RESUB is an 8K NOVA computer built by Data General with a standard teletype interface. It is advisable to have at least a 12K core.

RESUB will function as a stand alone program. Externals are checked before they are used.

Catalog No. 220-023

(LP)

SINGLE USER BASIC WITH MATRICES AND STRINGS

This version of BASIC is a modification of Data General's Time Sharing BASIC as described in the pamphlet called, "NOVA Line Time Sharing BASIC." The communications multiplexor handler has been removed and replaced with a handler that operates the console teletype only.

The Time Sharing Basic pamphlet should be consulted for the operation of this version of BASIC. It gives the operator all the program control commands described on page 3 of the pamphlet. An additional command (control P) can be used instead of a carriage return to have a listing made to the line printer, i.e., LIST P, or the results could be put to the line printer, i.e., RUN P.

Catalog No. 220-028

(LP)

MACHINE LANGUAGE I/O SUBROUTINES FOR T. S. BASIC

These subroutines allow TIME-SHARED BASIC users to manipulate data in binary or other arbitrary formats such as EIA, and to read and punch character string data using the high-speed reader and punch or a teletype reader/punch.

Manipulation of binary or other data is achieved by unpacking 8-bit characters into equivalent strings of ASCII 1's and 0's which can be manipulated in BASIC. Results can be repacked for compact storage or output.

The machine-language subroutines are accessed using CALL statements in BASIC. A version of TIME-SHARED BASIC incorporating these subroutines is available; it occupies about 150 extra words of core, but is otherwise identical to normal TIME-SHARED BASIC.

The supplied program includes the modified DGC Time-Shared BASIC (091-000026) with the machine language subroutines.

Catalog No. 220-031

NOVA OPERATING SYSTEM PROGRAM

This program, written for the Data General NOVA series of minicomputers, is used along with other user programs to list, enter, or fill core, and to read, punch, copy, or verify paper tapes. It can also jump completely out of itself to a user program somewhere else in core. The eight available commands provide for specified core dump, core modification (individual and block), core to binary format punch, tape copy and verify, core compare, and program start-up. Program resides in the 1K below Binary Loader. Minimum configuration is CPU with 4K of memory, Teletype ASR, and/or high speed paper tape reader. An 8K system is provided; changing 3 instructions in the source tape configures the program to any core size.

Catalog No. 220-033

EMUL-8

This emulator of a PDP-8/S/L/I computer runs on any NOVA with 8K of memory (min. 5K). Most instructions are emulated in 20-25 US. on a SUPERNOVA computer. The Emulator is organized so that the operator can carry out all control functions required for a PDP-8 from the NOVA console. In particular, it is possible to stop a PDP-8 program.

Allowance has been made for the special ASR 33 Teletypes used on PDP-8 computers. The address space is 4K 12 bit words.

This version was modified extensively to adopt fast routines used in SIMUL-8, while retaining the machine-like operation and the fast interrupt scheme of earlier EMUL-8's.

Catalog No. 220-035

CORE COMPARE AND LOAD

This is a program which compares absolute binary paper tapes with the contents in core locations. Any differences are printed out:

address	core value	tape value
---------	------------	------------

This version also corrects any differences; in other words, the core values are restored to the values on tape.

Catalog No. 220-036

BINARY TAPE MAPPER

The Binary Tape Mapper will print the starting and finishing addresses of all non-contiguous blocks on the tape. The end block address is also printed.

Note: Only tapes without "fill" blocks are read by this program. (Compatible with Vers. 1 of Loader).

Catalog No. 220-038

BINARY TAPE MERGER

The Binary Tape Merger Program allows the merging of a number of absolute binary tapes into one composite absolute binary tape. The output tape is punched in increasing address order and contains data for all those addresses defined in the input tapes.

Catalog No. 220-041

(LP)

COMBO BINARY RELOCATABLE EDITOR

COMBO facilitates editing and assembly of programs on NOVA systems with at least 8K of core. This program contains modified versions of the Extended Assembler and Editor programs. Additional Editor commands have been incorporated to control the Assembler as well as I/O device selection. Assembler input may be either from the Edit Buffer or from paper tape. The user determines the symbol table size at startup. Additional features make it possible to punch library tapes and to obtain partial listings.

Minimum peripheral required is Teletype, but COMBO will support high-speed paper tape reader and punch as well as line printer.

Catalog No. 220-042

FAST SPECIAL FORMAT FLOATING POINT SUBROUTINES

These nine subroutines were developed for fast real time applications. They include:

- FFLT Fix to floating convert
- FFIX Floating to fix convert
- FMOV Move FLPT number from one location to another
- FSR Store FLPT number (out of register) relative to AC2
- FLRS Load FLPT register relative to AC2 and skip if FLPT number is zero
- FADD Addition of two floating point numbers
- FSUB FLPT subtraction
- FMUL FLPT multiplication
- FDIV FLPT division

The program uses 713_8 locations and requires hardware multiply/divide. A short Basic program is included for putting constant floating point data into an assembler source program to calculate the mantissa and exponent out of a given value.

Catalog No. 220-043

INCREMENTAL PLOTTER SUBROUTINES FOR BASIC

This program extends BASIC with plotting capabilities by use of the CALL statement. Six calls are available: program initialization, XY-movements both absolute relative, pen up/down, ASCII character drawing, and coordinate inquiry. Program loading is described in the Extended Basic Manual.

Two sets of tapes are available with this program:

REV 05 - for use with Single User Basic with Call (user's Group Program no. 220-002)

REV 06 - for Extended Basic and Time Share Basic

When ordering, please specify which REV you want.

Catalog No. 220-048

MEMORY TEST PROGRAM

The Memory Test Program provides a comprehensive test of every location, excluding that required by the program, in a Data General read/write memory system. Every location is tested by using every possible bit pattern in that location. Under switch register control one has the option of using only a rotating bit pattern for each location, as well as the option of continuous cycling through the program. Errors are indicated by a halt, with the accumulators containing pertinent error information.

Any Data General processor with read/write memory is sufficient for operation.

Catalog No. 220-049

ASSEMBLER SOURCE PROGRAM ANALYZER

This ALGOL program scans an assembler source program and tabulates the number of instructions of each NOVA instruction type used. Group percentage usage as well as actual instruction type count is given as output.

Operating Instructions - Load the ASCII tape and compile the program according to the supporting operating systems procedures. After the object is loaded and started, the filename of the assembler program to be analyzed, including any extensions, is requested. After the program analyzes the source file, the statistics are printed.

Any ALGOL supportable memory is sufficient for program operation, and it will run under DOS, RDOS, or SOS.

Catalog No. 220-055

FLOATING POINT SORT

This is a callable, relocatable overlay-able subroutine written in FORTRAN. Its function is to take a tag file loaded onto any disk by the User and arrange that tag file in either ascending or descending order. The ordered information is left on the input file.

User must provide the mainline program.

Catalog No. 220-068

ERROR REPORTER

This module contains the main system error reporting routines. The errors are considered either fatal or non-fatal, and may involve a filename or not. If a fatal entry is used, a '.RTN' system call is issued after the error message is printed. A non-fatal error entry returns to the location following the error call, after the error message is issued. If a filename is assumed, the byte pointer must be in AC0 on entry to the fatal or non-fatal file error routine. The error module entry points are:

- 1) ERR1 - Fatal file entry point.
- 2) ERR2 - Non-fatal file entry point.
- 3) ERR3 - Fatal non-file entry point.
- 4) ERR4 - Non-fatal non-file entry point.

The error routines are intended to be called in the event of a system call error return. Example:

```
LDA 0, TEXTP      Try to open channel zero.
SUBO 1,1

.SYSTEM          Big SYSTEM CALL . . .
.OPENO

JSR .ERR1        Assumed to be a fatal error involving a
. . . .         file name.

.EXTN ERR1
.ERR1:ERR1      Fatal file error is declared external normal.
```

Supporting routines in this module are:

- OUT - Output a single byte to the current console.
- PRINT - Output text to the console (JSR + 1 = byte pointer).
- PRNT1 - Print text to the console (AC1 = byte pointer).
- OCTAL - Output a five digit value (AC1 = value) to console.

Each of the programs has been declared in an entry statement so that other programs may utilize them. All segments of the module are completely self-contained and do not call on any external application routines.

Catalog No. 220-069

DISK COPY

This is a standalone program for D. G. C. 203-cylinder moving-head disks of the type 4047A, B. 4048A, and 4057A. The program does a complete disk copy, all sectors. The time required is about 90 seconds for the 4047A, B; and 7 minutes for the 4057A.

Error messages are provided. The program will also run in a shared disk environment. Upon completion, the drives are recalibrated so that it is unnecessary to manually recalibrate for a boot.

DISK COPY is written in ASSEMBLY language.

Catalog No. 220-071

NOVA ACCOUNTING SYSTEM

This accounting system consists of a pair of programs capable of updating and editing a log file kept on a disk attached to a NOVA computer with at least 16K of core and RDOS. The system is primarily designed for machines operating in single-user mode, but is compatible with Data General Batch and Time Share Basic. Each user is known by name and four digit number, and must specify these correctly to gain access to the CLI. At the end of each session, the user calls the log program, which computes his total time used and then places the machine in an idle state. The source code for the programs is in FORTRAN IV, and locations for commonly made alternations have been marked with comments. The contents of the log file can be displayed at the console at any time by logging on with a special user name.

Catalog No. 220-080

PRINTX

PRINTX is a utility program which accomplishes the function of the RDOS "PRINT" command and, in addition, labels each printout with a header block containing the following:

- 1) Name of the file being printed
- 2) Current DATE
- 3) Current time
- 4) Current disk directory
- 5) Disk Pack identification

This is a convenient way of keeping track of the disk pack and directory in which a certain file resides.

PRINTX is used in the same way as the RDOS "PRINT" command, and runs under RDOS Rev. 3 in either foreground or background

Catalog No. 220-082

GENERAL PURPOSE FORTRAN MACROGENERATOR

The general purpose macrogenerator is a string processor in that it takes as its input a character stream and produces a character stream as output. The output stream contains strings directly copied from the input, and strings produced as the result of macro calls.

This macro processor will process arbitrary text strings. They need not be legal input to the Data General Macro Assembler. Features include recursive calls and definition and redefinition of macros within macros. One example of the program's usefulness would be for generating tables and test data during program debug.

Catalog No. 220-085

CASSETTE/MAG TAPE REVERSIBLE READ SUBROUTINES

These are a set of FORTRAN-callable subroutines which will enable a calling program to read a cassette, mag tape, or disc file in either direction in binary. The calling program must open the file, and provide a small portion of code which will rewind the file when returned to. Paper tape synchronization facilities are also included (sync flag is two completely punched frames) and the subroutines can even be used to "reversibly" read paper tape if the operator can re-load the tape each time a "rewind" is required. End of data can be sensed optionally by the detection of a zero datum word, or, of course, by detection of end-of-file.

Minimum configuration is a NOVA-line CPU supporting RDOS or SOS FORTRAN with appropriate peripherals to process the reversible file being read. Memory requirements will vary with size of calling program.

Catalog No. 220-088

RTOS SNAPSHOT DUMP

SNAP is a snapshot dump that was written for RTOS Rev. 5. SNAP outputs the contents of the accumulators, the carry bit, important page zero information (current TCB address, system mode indicator, current system interrupt mask, user stack pointer), all TCB queues with each task's priority, status, system call word, and identifier, and then returns to the caller.

Interrupts are disabled upon entry to SNAP and are not enabled until control is passed back to the calling program. SNAP does not restore the contents of the accumulators to their values upon entry to SNAP.

To use SNAP it must be included as an external normal. (.EXTN SNAP), redefined with some variable. (.SNAP: SNAP), and then perform a subroutine jump to it (JSR @.SNAP). The relocatable binary SNAP. RB is then included in the command line of the relocatable load.

NOTE: SNAP must be assembled with PARR.SR (ASM PARR/S SNAP)

SNAP is written in ASSEMBLY language.

Catalog No. 220-089

TASK - S SIMPLE MULTI-PROGRAMMING SUPERVISOR

Any number of parallel-running tasks may be defined and run under the supervision of the TASK supervisory module. The design objective was to allow programs to be written using the basic I/O facilities of the NOVA (DOAS, NIOC instructions etc.) in a shared environment with other I/O driving tasks. Control is given to the task of highest priority that is not waiting for an event. No time-slicing is used; the interval timer is not required. No device drivers are directly supported, since it is expected that the tasks which run with the TASK module will in fact be doing direct I/O. The TASK module traps interrupts, and enables any tasks which are waiting on the interrupt. A POST facility allows the use of pseudo-interrupts. One may use this to simulate an interrupt on any "event", such as a task completion.

TASK by itself requires 243₈ NREL locations and 5 ZREL locations. It also uses auto-incrementing location 20. It is written in ASSEMBLY language.

Catalog No. 220-090

BASIC PRINTING UTILITIES

This package of routines allows the Extended BASIC user considerable flexibility in listing ASCII files. PPT. SR and its two associated programs, PRTXREF. SR and PRTMAP. SR, are written in Extended BASIC to allow a user to:

- 1) List ASCII source and data files to any device or to the user's console with optional page headings and/or line numbers.
- 2) Copy sequentially organized ASCII files.
- 3) Obtain statement cross-reference and/or variable map listings for a BASIC source program file.

These programs require both RDOS and Extended BASIC Rev. 03 for operation.

STRIP (STRING PROCESSOR)

'STRIP' (String Processor) is a text-processing system for the NOVA family computers. In the 'STRIP' language, one can write procedures for accepting, naming and storing any character string from the input device; for modifying any string in any way; for treating any string at any time as an executable procedure, or as a name, or as text; and for outputting any string. The 'STRIP' language is based upon an extension and generalization to character strings of the programming concept of the "Macro". Through the ability of 'STRIP' to accept and store definitions of procedures, the capabilities of the language can be indefinitely extended. 'STRIP' can handle iterative and recursive procedures and can deal with character strings, integers, and boolean vector variables.

'STRIP' is written in ASSEMBLY language and requires a minimum RDOS configuration for operation.

Catalog No. 220-097

MUMPS

MUMPS is an interactive language and operating system that is very useful in dealing with large textual data bases. String handling is easy and efficient with the various string manipulation language elements. The hierarchical array and file structure provides easy organization of data. The language also has basic arithmetic capability.

This version provides the user with a basic MUMPS system. It includes almost all MUMPS language features and basic integer arithmetic capability. The system will run stand-alone and can provide a multi-user environment. 12K of core is necessary for a single user system; an increase in the number of users is a function of core available.

For simple systems with CPU, fixed head disk and teletype or CRT, very little tailoring is necessary. However, for larger and more complex systems, some software/hardware tailoring may be required.

MUMPS is supplied as an absolute binary tape with a manual and assembled source listings.

Catalog No. 220-100

RDOS/FORTRAN IV I/O SUBROUTINES

This package contains six subroutines that allow FORTRAN IV programs to use RDOS character, sequential binary and ASCII line modes of I/O without the requirement of using the FORTRAN formatted I/O package. The subroutines complete the FORTRAN IV interface to RDOS that previously only included record and block modes of disk I/O and free formatted mag tape I/O. The routines included are: character I/O, sequential I/O and line I/O subroutines, and they are written in ASSEMBLY language.

CHARACTER PROCESSING ROUTINES

This package of FORTRAN routines consists of the following:

- INDEX - will search the specified elements of an integer array for a particular word. If the word is not found, the function returns a value of zero (0). If the word is found, the function returns the location of the word as its value.
- IFIND - will search the specified elements of an integer array for a particular string. If the string is not found, the function returns a value of zero (0). If the string is found, the function returns the location of the first position of the string in the integer array as its value.
- RFIELD - will right-justify (with leading blanks) characters in a field. Imbedded blanks will be retained in the field.
- IBYTE - will pick-up a byte from a string array.

The routines are an extension to the Data General Commercial Subroutine Package, and require the FORTRAN Runtime Library for operation.

Catalog No. 220-102

RTOS/RDOS SYSTEM UTILIZATION TASK

The System Utilization task "UTIL" allows the user to monitor the amount of CPU time not used by the application program and the operating system. It operates with either RTOS or RDOS and utilizes the multi-tasking capability of the systems. Once a minute the task "UTIL" will output, to a user specified file or device, a message of the following format:

MM:SS idle time = HH. TT%

where MM:SS is the current time in minutes and seconds
HH. TT is the percentage of CPU time that is not being used by the operating systems or user application tasks.

The program is written in ASSEMBLY language.

Catalog No. 220-103

DTECT

This routine can be used to monitor the line activity on all unopened QTY lines. If QTY:64 is opened on the supplied channel and this routine is called, it will return after sensing activity on any unopened lines. Upon return, "Line" will contain the line number where the activity took place, and "Noise" will contain the character that was entered.

DTECT is written in ASSEMBLY language and requires the FORTRAN IV Runtime Library.

Catalog No. 220-104

CALENDAR

This Fortran routine will get the time of day and the date and format them into ASCII strings for output. Example:

10:10 AM Wednesday June 26, 1974

The FORTRAN Runtime Library is necessary for execution.

Catalog No. 220-105

F1130 - IBM 1130 FORTRAN PROGRAM CONVERTER

This program converts IBM 1130 FORTRAN programs (source level) to DGC FORTRAN source code.

The program performs the following checks and conversions:

- a) Converts (Label) Read (CHL, RECNO) List
to (Label) Call FSEEK (CHL, RECNO)
Read Binary (CHL) List
- b) Converts (Label) Write (CHL, RECNO) List
to (Label) Call FSEEK (CHL, RECNO)
Write Binary (CHL) List
- c) Converts (Label) Find (CHL, RECNO)
to (Label) Call FSEEK (CHL, RECNO)
- d) Does the following subroutine calling sequence changes:
Call ADD (...) To Call QADD (...)
Call SUB (...) To Call QSUB (...)
Call MPY (...) To Call QMPY (...)
Call DIV (...) To Call QDIV (...)
Call MOVE (...) To Call QMOVE (...)
Call READ (...) To Call QREAD (...)
- e) Converts (Label) Call LINK (Name)
to (Label) Call CHAIN
("NAME.SV", IER)
X Type 'CHAIN ERROR =',
IER
X Stop
- f) Puts all variables initialized by a data statement in labelled common via the following statement
"COMMON /LBCOM/ ... /

g) Places comments at appropriate places in the program text to make the program more readable.

h) Optionally converts all 026 keypunch codes to 029 codes.

& = +
% = (
15392 =)
= =
@ = '

i) Converts the following numeric code representations:

5440 = 3360
16448 = 8224
24896 = 12064
11552 = 24640

j) Converts (Label) Call P1403 (...)
(Label) Call PRINT (...)

k) Converts (Label) Call S1403 (...)
Call S1403 (...)
(Label) Continue

l) Does not insure that variables equivalenced together exist in common.

Catalog No. 220-106

DISASSEMBLER

The Disassembler is a 2 pass program that reads in an absolute binary file and produces the appropriate assembly language. The program will produce symbolic references (in the form AXXX, where XXX is a 4 digit octal number) for all memory reference instructions. This requires that the binary file be read twice.

At assembly time, one must specify if the program is to be run standalone, under DOS or under RDOS. With standalone mode, output is a listing on the line printer and an optional source tape in the teletype punch. Input is from the high speed paper tape reader. (A user written print routine is necessary to accept a character and print it in this mode).

For DOS or RDOS, input may be from any file, and output is either a listing or a source file (but not both), depending on whether the /A global switch is present. Either the source file from the standalone or operating system version may then be input to the assembler.

Output consists of the instruction in ASSEMBLY language, the instruction decoded to 2 ASCII characters (if printable ones) and for DOS or RDOS the appropriate .SYSTM call word if it could be interpreted as such.

Several unfamiliar instructions such as NIO 2, LPT are produced but the appropriate .DIOA instructions are appended to the source file to handle these instructions when they are input to the assembler. The Disassembler requires about 3700 (octal) core locations and sufficient core storage for the number of memory reference instructions that reference different addresses. (A minimum of 4K is suggested for the standalone version).

The program is written in ASSEMBLY language.

Catalog No. 220-107

MCA BINARY LOADER

The MCA loader allows an absolute binary program to be loaded via the MCA (Multiprocessor Communications Adapter no.4038). One computer must be running with unmapped RDOS. The program will load any RDOS file (disk or paper tape). The receive routine is a modified binary loader which when loaded replaces the standard loader at the top end of memory. The modified binary loader will either accept input from the high speed reader or from the MCA. This program is about 150 words larger than the standard binary loader, and utility programs such as the assembler and BASIC may overwrite a section of this loader necessitating the reloading of it before it can be used again. Once the receive routine is ready (in execution) the binary file can then be loaded via the MCA.

Note: The program as now written, uses MCA receiver 1 and 2. These two receiver codes are noted in the program and must be changed if the MCA that is being used does utilize these two receiver codes.

The program is written in ASSEMBLY language.

Catalog No. 220-108

FILECOMPARE, BINARY-PACK

This program is similar to the RDOS CLI FILCOM command except that it does a byte by byte comparison (instead of a word) of two files. Also leading nulls on one or both of the input files are skipped. This allows a file on disk to be compared to a file on paper tape that has leader on it. Dissimilar bytes are printed either to the system console. (\$TTO) or to a listing file, if specified. The displacement into the field where the difference occurred is also printed.

FILECOM is written in ASSEMBLY language.

Catalog No. 220-109

NOVA INTERPRETER

This program interprets another ASSEMBLY language program insuring that users do not attempt to reference or execute outside of their area of core. (The valid region of core is established at load time). The program will catch the following user errors:

1) Not starting address.

- 2) Attempt to execute outside area.
- 3) Attempt to reference outside area.
- 4) Too many levels of indirect addressing.
- 5) Attempt to branch to itself.
- 6) Execute time exceeded.
- 7) System call not allowed.
- 8) Direct I/O not allowed.
- 9) Job cancelled by operator.

If any of these error conditions occur, program execution is terminated and a dump is produced. Execution of a 'HALT' instruction causes program execution to terminate normally and to produce a post execution dump.

Either at assembly time of this routine or during program execution, I/O instructions and/or .SYSTEM calls can be made valid.

Tracing of up to 10 statements or complete sections of a program is provided. At any time during program execution, a dump of the user area or just of the accumulators may be requested.

The requirement for execution is that the user program be relocatable and have the symbol 'START' declared as an entry point (using the .ENT statement) and that 'START' be defined on the statement where program execution is to begin.

The program is written in ASSEMBLY language and requires about 2000 octal words of core. It has been used in a student environment to prevent inexperienced users from storing over part of the operating system or branching off wildly.

Catalog No. 220-110

FORTTRAN RANDOM NUMBER GENERATOR

This program is a function subprogram that is callable from a FORTRAN program, and will generate a sequence of pseudo numbers in the range from 0.0 to 1.0 (0.0 = X 1.0). The program has two entry points, RAND and RND, which produce the same results. In the first call to this routine the argument given in the call is the initial seed for the sequence of numbers to be generated. A sequence is repeatable if the same seed is given in the first call.

Note: The argument given must not be a constant because this routine changes its value every time that it is called.

The subroutine is written in ASSEMBLY language and requires 36 octal words of core.

Catalog No. 220-115

BASIC EDITOR

This program is written in Extended BASIC and it allows a user to edit programs in BASIC using either single or timesharing mode. It is primarily an implementation of a subset of the NOVA Text Editor features. MACRO capability is not present. The following commands are available:

- a) Insert following lines
- b) Type the lines
- c) Delete the lines
- d) Change the string to a new string
- e) Search for line containing string
- f) Specify input file
- g) Specify output file
- h) Yank a page
- i) Punch a page
- j) Current line position
- k) Total no. of lines
- l) Set line position to first line
- m) Set line position to after last line
- n) Append next page to buffer
- o) Close output file
- p) Type condition of the text buffer
- q) Return back to the monitor

Catalog No. 220-117

RDOS BINARY LOADER

This routine is a modification of Data General's "BINARY LOADER" (Document 093-000003-06) so that it can be loaded via RDOS HIPBOOT. This allows shifting from RDOS operation to a standalone one and having the binary loader already in core. It is loaded either with the CLI 'BOOT Filename' command or by responding to HIPBOOT 'FILENAME?' with the save file name. The routine sizes available core and relocates the binary loader to the high end of core.

Catalog No. 220-119

PREP

This program provides a facility which is often called the 'include feature'. In general, PREP scans a file containing source statements for statements of the form:

(SP) Include (SP) (FILENAMES)
(CR)

WHERE: (SP) Means any number of spaces
and/or tab characters

(FILENAMES) Means any number of
filenames, separated by
blanks, tab characters and/or
commas

(CR) Means carriage return
character

If such a statement is found, it is replaced by the contents of the successive files named on it. These contents, in their turn, are also inspected for the occurrence of include statements. The depth of nesting is four.

The main program is written in ALGOL. An ASSEMBLY language routine called ('FGRND'), which is used to investigate whether PREP is running in the background or foreground, is also included.

Catalog No. 220-123

FORTRAN POST PROCESSOR

This ALGOL program will perform the following modifications given an assembly source output from the FORTRAN IV compiler:

- a) All occurrences of:
JSR @.LDX (or .STX where X = 0, 1, or 2).
(ABS. ADDRESS)

Are changed to:

LDA @.+2
JMP .+2
(ABS. ADDRESS)

- b) All single dimension array subscript calculations are converted to 'inline' code.

Catalog No. 220-131

FOREGROUND CLI

The FOREGROUND COMMAND PROCESSOR (FCLI) is a CLI type utility program designed to run in the foreground of an unmapped RDOS machine. FCLI will work with any RDOS revision 3.00 or higher. FCLI occupies only 1436₈ words of NREL core and no page zero, so it can be left core resident at all times. FCLI communicates through the second teletype (\$TTI1/\$TTO1). If a second teletype is not present at the system the FCLI source could be modified to use the background console and operation messages.

Catalog No. 220-133

SYMBOLIC ALPHANUMERIC GENERATOR

This program punches 8 Bit High Characters (from ASCII 40 to ASCII 137) on the teletype punch. Characters are input via the TTY which does not echo input to the punch. This program is useful for labeling tapes and making signs (characters are on reverse of tape from manufacturer's label). Program is relocatable but the set may prefer to make it absolute and store it in some part of memory permanently.

Catalog No. 220-136

MYPRI

MYPRI is a subroutine which returns the task priority of the caller. It is provided in two versions -- FORTRAN IV callable and ASSEMBLY language callable.

Catalog No. 220-138

EBCDIC TO ASCII TRANSLATOR

These two programs are capable of reading an IBM mag-tape into a Data General disk file and then translating the disk from EBCDIC to ASCII code. The programs can be used to translate FORTRAN source decks obtained on magnetic tape in IBM format into ASCII for use on a Data General machine.

Catalog No. 220-139

PRNT - LISTING ROUTINE

This program allows a user to print a source program as a formatted listing with line numbers, page numbers, and creation date. It is similar to an ALGOL brief listing.

Catalog No. 220-140

ALGOL EDIT AND DUMP PACKAGE

Three programs are provided:

- BEDIT - Similar to OEDIT except that it provides editing in any radix.
- HDUMP - dumps a file to the line printer in hexadecimal format.
- BDUMP - dumps a binary file to the line printer in any radix.

Catalog No. 220-141

BASIC LINE EDITOR

A BASIC program which allows line editing of non-basic ASCII files using six commands: add, insert, delete, change, print to user terminal, and print to line printer.

Catalog No. 220-145

FORTRAN IV GET CHANNEL ROUTINE

This ASSEMBLY language routine is a subroutine that is callable from a FORTRAN IV program to obtain a free FORTRAN channel number. This routine is especially useful in a multi-tasking environment, as specific channels do not have to be decided upon, but can be determined at execution time.

Catalog No. 220-146

UNICO

UNICO is an ASSEMBLY language program which was developed for converting a file created with 029 punched cards to a file suitable for transmission to a Univac machine using DG's RSTCP.

This program is extremely useful to any user who previously used an IBM 1130 to emulate a Univac 1004 terminal and who can use RSTCP to communicate with a Univac service bureau supporting 2780 terminals.

Catalog No. 220-147

IBMCO

IBMCO is an ASSEMBLY language program which produces a 9-track magnetic tape capable of being read on a 9-track IBM tape drive. Input to the program is an ASCII file with a maximum record length of 80 bytes stored one character per byte. The ASCII characters are converted to an Octal code which, when read on IBM tape drive, will have the proper hexadecimal code to represent the original character. The IBM utility named DITTO is used to retrieve information from the free format tape and transfer it to an IBM output device.

Catalog No. 220-149

CLEAN

CLEAN is a file maintenance program which lists all files last accessed on or before a cutoff date along with the last access date and the size in sectors and finally the total number of sectors (both in decimal) OCC by these files for a given directory. Printout can be limited to only 1 subpartition or subdirectory. If the delete feature is selected, all the files which are listed are also deleted.

Catalog No. 220-150

MAPCK

A standalone program to ferret out errors in SYS. DR and MAP. DR on either of 2 Diablo-type moving head disk cartridges (DP0 & DP1) with 203 tracks and 24 sectors/track. Starting address is 2.

Catalog No. 220-151

EIAAS

EIAAS is a FORTRAN IV callable ASSEMBLY language subroutine which will perform bidirectional conversions between ASCII code and EAI numerical control code. Input to the subroutine is the direction of conversion and a word packed with two characters to be converted. Output consists of the converted left and right bytes of the input word each left-justified in a separate word plus a word packed with both converted characters.

Catalog No. 220-152

FORTRAN ERROR MESSAGE INTERCEPTOR

This routine is called from a FORTRAN IV program and allows error messages that may be generated during the execution of the program to be output to a user selected file and also to be output as usual to the system console.

Catalog No. 220-153

DIRECTORY LISTING PROGRAM

This program is used to list all directories and devices that are initialized in an RDOS Rev. 4 system. It is especially useful when the system reports that no more DCB's are available. A list of initialized directories and devices can then be obtained and those not needed can then be released freeing the DCB for further use.

L. M. U. PACKAGE

Language: BASIC

The computer department at Loyola Marymount University has contributed its library of programs to the User's Group. More than one hundred routines of varying interest are included in the package. The entire package must be ordered, as **individual programs are not available**.

NOTE: These routines were written under BASIC, Revision 3, and will require some changes to be compatible with later revisions of BASIC.

AGEBLEND	An application of the Meyers Method of "blending" to a single year of age
ANNUIT	Performs calculations for both payment and withdrawal annuities
ANVAR 2	Computes an analysis of variance table and F-ratios for a simple Latin square design
ANVARI	Computes an analysis of variance table for a completely randomized one-way design
ASGNMT	Solves the classic assignment problems and determines the cost of implementing the solution
BAIRSTOW	Finds the roots of polynomials using Bairstow's Method
BALANCE	Calculates past and projects future population growths by solving for all components of the "demographic balancing equation"
BEAM	Recommends the correct steel beam to use for a number of common applications
BICONF	Calculates the confidence limits for a population proportion based on the exact binomial distribution
BINDIF	Computes the binomial distribution function for problems of the form
BINOPO	Compares the exact binomial probabilities with approximations given by the normal and Poisson distribution
BITEST	Makes the necessary calculation for a statistical test of a binomial proportion
BLKJKSZ	Blackjack game
BODE	Prepares table of Bode plot data
CALL	Relays messages from the user to the computer operator

CDETER	Computes value of complex determinant using the Crout Method
CHI	Applies chi-square test to several sample proportions
CHISQS	Computes chi-square statistics for any number of M*N contingency tables
CIRCLE	Divides circles into any number of equal parts giving the angle in decimal degrees and degrees, minutes, and seconds, and also the horizontal and vertical distances from the center to the point on the circumference
CONBI	Confidence limits for a population proportion based on the normal curve approximation
CONDIF	Computes confidence limits for the difference between two population means
CONLIF	Computes confidence limits for an unknown population mean based on random sample data given
CRAPS	Simulated crap game
CRIT	Solves project scheduling problems which are formulated in terms of a network
CROUT1	Solves linear equations using the Crout Algorithm
CTRFFT	Complex to real fast fourier transform
CURFIT	Performs a least square curve fit
DDEPRE	Calculates annual depreciation by using the sum-of-the-digits method
DE10R	Solves the initial value problem for a first order differential equation by the second order Runge-Kutta Method
DE20R	Solves the initial value problem for a second order differential equation by the second order Runge-Kutta Method
DECIDE	A decision tree analysis model
DECSN	Furnishes the simulated business conditions and the mechanics for operating a business game for 10 to 60 participants
DEFOG	An instructional management simulation game
DERIV	Gives approximation to the derivative of the function FNF (x), at the point x = a
DERTIN	Computes a set of approximations to the derivative of any function F (x), for any value assigned to the independent variable x
DIFFEQ	Solves fourth order Runge-Kutta-Gill integration problems

DISSIM	Measures the degree to which a subgroup of a population is segregated in residential areas apart from the remainder of a city's population	LCVSIL	Calculates liquid control valve coefficients and determines required valve rangeability
EIGJAC	Eigenvalues and Eigenvectors of real symmetrical matrices by Jacobi	LENDER	Calculates the monthly interest charges and outstanding balance of a loan that must be paid off in a year or less
EQUITY	Computes the cost of equity capital by computing dividends and the share price for future periods (based on the Gordon model), and then finds the discount rate by equating the present value of the stream to the current share price	LIFENCHS	Constructs an abridged life table from age-specific death rates and conversion factors using NCHS method
EUPHORIA	The Kingdom of Euphoria game which simulates building wealth and power in a feudal society	LINFIT	Computes the best linear fit and correlation for a set of independent variables to a dependent variable
EXTRAP	Estimates populations by the extrapolation of two known censuses	LINPRO	Maximizes an objective function using the two-phase method
FRIKT	Computation of friction and energy	LNREG1, LNREG2	Compute slope and Y-intercept for various linear regressions
FUTBAL	Football game	LNTRND	Computes a simple linear trend forecast with seasonal adjustments for monthly data
FUTURE	Statistical forecasting model that computes forecasts by the statistical methods of mean, regression line, seasonal averages, seasonal regression line, moving average and exponential smoothing	LPFILT	Designs low pass filters
GAS	Calculates gas and vapor (but not steam) control valve coefficients and allows you to determine required valve rangeability	MIS	Calculates Eigenvalues and Eigenvectors of an N-by-N real matrix
GEOMEN	Computes the geometric mean and standard deviation for geometrically normal set of data	MOMEN	Computes data for momentum force curves at bridges for rectangular or trapezoidal channels
GFFT	Finds the fourier transform of a function using the Cooley-Tukey Algorithm	MORTGAGE	Creates monthly and yearly tables for home mortgages
GLSPA1	Solves linear programming problems via the two-phase simplex method	NATINC, NATINC3	Implements simple models of the national income
GMCRO1, GMCRO2	Fiscal policy game	NETWK	Hardy Cross method of pipe network analysis
GMCRO5, GMCRO6	Economic policy game	ONE	A simple population model
GSLDFL	Gauss-Seidel solution of load/flow problems	ONEWAY	Performs a one-way analysis of variance with equal sample sizes
HALF	Calculates the zero of a function by the bisection or half-interval method	PHICOE	Computes the value of the phi-coefficient and the number of cases for data on two variables
HISTOG	Prints a normalized histogram from a set of data	PLOT	Plots single-valued functions of x with x on the vertical axis
INSYS	Computes inventory levels and factory output for a factory-wholesaler-retailer inventory system	POLYFIT	Fits least-square polynomials to bivariate data using an orthogonal polynomial method
INTFCS	Evaluate a definite integral by Simpson's Method	PYRAMID	Constructs age-sex pyramids from demographic data
INTPTS	Performs numerical integrations of function described by (up to) 100 data points	RANGE	Simulates the results of firing a field artillery weapon
LADDER	Network analysis for circuits with a ladder topology	REGCOR	Performs simple regression and correlation analyses on a series of observations of the values of the two variables

RK2	Solves second order differential equation by fourth order Runge-Kutta Method	TAPE	Creates a tape which is readable by the high speed paper tape reader
ROMINT	Integrates a given function by the Romberg Method	TCPLOT	Plots data on either a linear or logarithmic scale
ROULETZ	A roulette game	TESTUD	Performs calculations necessary for testing an unknown population mean using sample statistics
SAVIN	Calculates the amount of money that accumulates in a savings account	THREE	Population model in which the rate of change of the population is assumed to vary as the square of the population
SCHUL	Tabulates and plots the results of a school test	TMFCEZ	Evaluates time functions which are sums of exponentials and exponential sine-cosine terms
SIMEQ	Solves systems consisting of n linear equations with n unknowns	TRIANG	Finds the unknown features of any triangle given one side and any two other parts
SIMPLE	Automatic simplex linear programming routine	TRUINT	Calculates the true annual interest rate charged on an installment loan
SIXPLOT	Plots 1 to 6 functions simultaneously	TWO	Population model in which the rate of change of the population is assumed to vary as a function of the current population
SMA	Samuelson's multiplier-accelerator model	TWO PLOT	Simultaneously plots two functions of a single variable
SMLDRS	Calculates the mean, standard deviation, and standard error, and plots a histogram for up to 300 pieces of data	XY PLOT	Plots single-valued functions
SORT	Sorting program for records with a standard length in ascending or descending order	ZEROES	Locates the values of x at which relative maximums and minimums of F(x) occur, and the values of x for which F(x) is zero (i.e., the zeroes or roots of the function).
SQC	Statistical quality control model that computes the control charts for the mean and range for a group of sample data	ZGEN	Generation of ZBUS matrix from primitive impedance matrix
SQRZ	Computes square root of a complex number		
STARTREK	Version of the Startrek game (based on the T. V. series of the same name)		
STAT1	Computes the mean difference, variance of distance, standard error of difference, and T-ratio for two groups of data, unpaired		
STAT2	Compares two groups of data by means of the Mann-Whitney two sample rank test		
STATAN	Computes 34 different measures for an array of weighted or unweighted values of the variable and also gives a 10-class frequency distribution summary		
STEAM	Calculates steam control valve coefficients and determines required valve rangeability		
SYLOG	Computes a Boolean expression to yield either an evaluation of a specific case of the expression, or a general truth table for all possible cases		

Catalog No. 220-155

BRTIOS (A-D I/O and Logical Functions for the BASIC User)

BRTIOS uses the CALL features of Extended BASIC to provide analog and digital I/O and logical functions to the BASIC user. The package supports the 4120 Series A/D, 4180 Series D/A and 4066 Digital I/O. It is supplied as a source file utilizing CALL numbers 10 - 19 and must be assembled and SYSGENed into Extended BASIC following the instructions in the manuals (93-65 and 93-119). The 10 SUBROUTINES ARE: Analog Input, Analog Output, Digital Input, Digital Output, Set Bit, Clear Bit, Get Bit Status, Exclusive Or, Inclusive Or and Logical And.

Catalog No. 220-157

PAGE

This program will paginate program listings on a Centronics Series 100 matrix line printer. The only requirement for this program to operate is that the program to be listed must have previously been listed to an ASCII file. The program will list your program in 55 line pages, print the title of your program in double width characters at the top center of the page, number the pages, and indent for punching notebook holes in your listing. It also takes care of lines that are larger than the page width.

Catalog No. 220-159

PROGRAM SPACE MEMORY SIZER

This program is written to reduce the tedium of determining program memory requirements in unmapped foregrounds, but it is also useful in other environments. It is run under any **Rev. 3 or greater** of any RDOS system, in either partition. The only I/O device used is the partition console to print results. If an error is encountered a "?" is printed followed by a CLI error message. This program occupies 1K of memory.

Catalog No. 220-161

CRT TO DISC DATA FORMATTER

This program simulates the drumcard feature of an IBM keypunch and produces line image ASCII data files. User control or automatic tabbing and duplication are provided. The user is directed to set up the designed "drumcard" prior to entering his data or to select from a number of frequently used drumcards which can be defined in the source program. The current version contains a Cobol source and an MPSX matrix drumcard. The program is written in ALGOL and requires an RDOS system with at least 16K words for compiling the ALGOL source, 12K words to run the program, a CRT, and a line printer (optional).

Catalog No. 220-163

FORTTRAN IV COM. CM READ ROUTINE

This is a subroutine which is callable from a FORTRAN IV program to read the contents of the CLI communications file COM. CM for background execution and from FCOM. CM for foreground execution. On the first call to this routine, the file will be opened and it will return the program name and global switches. Successive calls will return any parameters that were entered with the CLI command that invoked the program and also any local switches.

Catalog No. 220-165

EX UTILITY

EX is a RDOS utility program to perform a FORTRAN IV, ALGOL, relocatable assembler or Macro assembler compilation, loading, execution and debugging of one or more source files. The output includes one or more relocatable binary files and an executable save file. EX can treat source files individually where some require loading, others (macro) assembly and loading, and still others compilation as well.

Catalog No. 220-199

CHAT

The CHAT program may be used to change the attributes of all files in the current disk directory, either to permanent or non-permanent. It has the VERIFY and CONFIRM options as does the CLI DELETE command.

Catalog No. 220-200

COMPILE

The COMPILE program enables the FORTRAN compilation of multiple FORTRAN source programs on one input file. Output is a library file named COMPILE.LB. Compilation may be either in FORTRAN IV or FORTRAN V. Usage instructions are given in the program listing.

Catalog No. 220-201

DISKBOOT

This program will boot up DP0 and is itself bootable from a magnetic tape device. For example: install the save file 0 of CT0 or MT0. Manually rewind the device. Set console switches to boot up the device. Hit RESET and PROGRAM LOAD. The device will boot, then rewind, then the disk will boot.

Catalog No. 220-202

SBRTB

The SBRTB program, when loaded with BASIC revision 4, permits the user to invoke any program on the disk. For example, the user may invoke the editor to do sophisticated editing of source files, or he may invoke the CLI for its features. To return to where he left off in BASIC, he does the normal exit from the invoked program (POP from CLI).

Catalog No. 220-203

TCOPY

The TCOPY program reads magnetic tape files in either DG or any other format. The input tape must be on drive-0. The output filename is specified in response to inquiry by the program. MT0 is initialized and released by the program.

Catalog No. 220-204

MTUTIL

MTUTIL performs a variety of magnetic tape operations on the Eclipse 9-track 800 bpi transport. It accepts commands from the console or a disc file; these commands may be used to write simple programs using a looping construct, operator intervention, etc. MTUTIL functions include: spacing, checking for various errors, dumping in various formats, and copying tapes via the disc.

Minimum configuration required: Any RDOS system 14K min. with a 9-track 800 bpi; transport and the standard FORTRAN V library.

Catalog No. 220-206

GANDER

GANDER is a key-to-disc data entry program written in AOS Extended Basic Rev. 1.00. AOS users should have no trouble bringing the program up on their system if they follow the instructions outlined in the implementation notes. If you want to run it under RDOS, Rev. 4.1 BASIC is necessary and minor modifications will have to be made to GANDER's source code (described in implementation notes). The tape is supplied in an RDOS dump format so that it can be loaded onto either RDOS or AOS (the latter by using the RDOS utility program).

Besides GANDER.SR (the basic source code), the tape also contains in file 0 the following:

GANDER.MA - A 57 page user manual

GANDER.IM - Implementation notes for AOS/RDOS

GANDER.DO - A documentation file containing brief descriptions of GANDER commands and operating procedures.

Catalog No. 220-209

ASM 16 (IMP-16 CROSS ASSEMBLER)

This FORTRAN package provides a 3-pass assembler for IMP-16 ASSEMBLY language routines. It produces a listable file of the assembler and an object file in paper tape (binary record) format.

Minimum configuration is a NOVA 40K RDOS Rev. 05 system with a CRT (or teletype) and a line printer as well as an input device (magnetic tape or paper tape) for the IMP source files.

In addition to the root program, there are 39 routines and the FORTRAN runtime library included on the ASM16 tape.

Catalog No. 220-210

SOURCE LEVEL SYMBOLIC DEBUGGER FOR ECLIPSE FORTRAN 5

With information provided by the Data General FORTRAN 5 compiler, this program is capable of performing the following functions:

1. Starting execution of the user's program and stop execution at dynamically user defined points in the program.
2. Monitor execution of a user's program.
3. Allow the user to inquire where control currently is in the program. The reply is made in source level terms (source code line number and routine name).
4. Allows the user to display the current values of variables and change their value by using the FORTRAN source variable name.

Minimum configuration: Any ECLIPSE that qualifies for FORTRAN 5 RDOS/INFOS.

Known restrictions: This program does not support multitasking under FORTRAN 5 Rev. 5.10 and up. FORTRAN 5 Rev. 5.20 deletes this deficiency.

Catalog No. 220-211

LC/50 INTERFACE

This program is a .IDEF driver for the Documentation LC50 Card Reader/Punch/Printer. The LC50 controller (5629), and the LC50 card handler (5629A) are sold by DGC special systems.

The tape consists of an exercise program that demonstrates the use of the driver, the driver, and the driver specifications.

Minimum configuration: Any RDOS system Rev. 5, 5629, 5629A.

Catalog No. 220-214

LISP 1.5 INTERPRETER

LISP is a high-level list-oriented symbolic programming language. It is a formal mathematical language, and as such can be described in a precise manner. The language has been used extensively in artificial intelligence research and many of the more modern languages in this area are based on LISP, and sometimes embedded in it.

This interactive LISP 1.5 interpreter has been written for the Data General NOVA line computers to run under RDOS. This version is generally a subset of the original IBM 7090 LISP 1.5 interpreter but does offer some additional features which make it quite a usable system. These features include the capability to read from multiple input files created with a text editor and the capability to provide graphical output to a Textronix 4010 terminal.

A complete documentation manual is included. The manual assumes that the reader has a working knowledge of LISP. Perhaps the best available introduction to LISP for those not familiar with the language is C. Weissman's *LISP 1.5 Primer* (Dickenson, Belmont, California, 1967).

Catalog No. 220-215

BASIC ACCOUNTING FILE SUMMARY

MUBAFSR is a BASIC program written in modular, top down, structured style that prints a summary report of the BASIC Accounting File by account numbers as specified in the BASIC ID file. The report includes the following information:

- account identifier
- allocated directory name
- month, day and year of the last activity
- total CPU time in seconds
- total I/O references
- total wall time in minutes

The program may be viewed as a kernel for the system manager referred to in the manual "NOVA Extended Basic System Manager's Guide (no. 093-119) where the BASIC AF file is described. Enhancements such as posting/billing by account number are easily incorporated due to the program's modular design.

Catalog No. 220-217

ALEX.AL

This program automatically compiles, loads, and executes ALGOL programs with external procedures, either in ALGOL or ASM. Compilation and loading are done conditionally, i.e. only after the source program has been updated.

Catalog No. 220-218

ALGLIB

This is an extended CFE program, which keeps track not only of the LB Library files, but also of a LF file which contains the source code of the modules in LB.

Catalog No. 220-219

ALGPROC

This is not a program leading to an SV file, but a collection of ALGOL procedures, either .AL or .SR. Several of these procedures are required in the ALEX, etc., ALGOL programs, but may be used as external procedures for other programs.

Catalog No. 220-220

SCRIBE

SCRIBE is an ALGOL program that processes text files and formats these files into pages. Output can be to TTY or to other files. Paper size, width, margins, etc. can be adjusted by commands included in the input text.

Catalog No. 220-221

TANKA

This program was written as a demonstration of string manipulation techniques. It prints poetry, interactively, in Japanese TANKA style. The program asks for two numbers and a command format file. (TANKA), the first number initializes the random number generation, the second number sets the number of stanzas produced. For any given order of word classes (found in TANKA, words stored in TANKA <0,1,2,3>). The program reads a random word and formats stanzas with specified numbers of syllables.



PROGRAMS LIBRARY PRICE LIST

Program No.	Name	Magnetic Tape (M)	Paper Tape (P)	
220-001	Fast Fourier Transform and Inverse	\$10	\$10	
220-003	TTY Tape Edit	5	5	(Listing Only)
220-004	BEAP Editor/Assembler	10	10	
220-005	EIGENVALUE Solution Program	5	5	(Listing Only)
220-010	Punch Card to Paper Tape Converter	NA	10	
220-011	NOVA Mag Tape I/O System	10	10	
220-012	Hollerith to ASCII Conversion Subroutine	10	10	
220-013	BCD to Binary Conversion by Radix Reduction	5	5	(Listing Only)
220-015	Pin List System	10	10	
220-017	I/O Package	10	10	
220-019	Wire List Layout	10	15	
220-020	Overlay Editor Assembly	10	10	
220-021	RESUB	10	10	
220-022	Focal Length by Microscope Method	10	10	
220-023	Single User BASIC with Matrices/Strings	10	10	
220-024	Conductor Selection for Low Voltage 3-Phase Systems	10	10	
220-025	Linear to Metric Conversion	10	10	
220-026	Metric to Linear Conversion	10	10	
220-027	BASIC Football/Baseball Games	10	10	(No Documentation)
220-028	Machine Language I/O for T/S BASIC	10	10	
220-031	NOVA Operating System	10	10	
220-033	EMUL-8	10	10	
220-035	Core Compare and Load	10	10	
220-036	Binary Tape Mapper	10	10	
220-037	TAD (Tape to A/D - D/A)	10	10	
220-038	Binary Tape Merger	10	10	
220-040	Polynomial Curve Fit (Polyfit)	10	10	
220-041	Combo Binary Relo Editor	10	10	
220-042	Special Floating Point	10	10	
220-043	Incremental Plotter Sub. for BASIC	10	10	
220-044	Curvefit	10	10	
220-045	Arctangent	10	10	
220-046	DACBAP	10	10	
220-048	Memory Test	10	10	
220-049	Assembler Source Analyzer	10	10	
220-050	Reaction Time Lab (RTLAB)	15	35	
220-052	Whole	10	10	(No Documentation)
220-053	Single/Double Precision BCD/BIN	10	10	
220-055	Floating Point Tag Sort	10	10	
220-056	RTOS 2702 Emulator (IBM)	10	15	
220-058	BATNUM	10	10	(No Documentation)
220-059	Slot	10	10	(No Documentation)

Program No.	Name	Magnetic Tape (M)	Paper Tape (P)	
220-062	Blackjack	\$10	\$10	(No Documentation)
220-064	Turtle Race	10	10	(No Documentaiton)
220-065	Lunar Lander	10	10	(No Documentation)
220-066	Horsrace	10	10	(No Documentation)
220-067	Heman	10	10	(No Documentation)
220-068	Error Reporter	10	10	
220-069	Disk Copy	10	10	
220-071	NOVA Accounting System	10	10	
220-072	2-Player Football	10	10	(No Documentation)
220-075	Acid Bace Titration	10	10	(No Documentation)
220-076	Class Grading Program	10	10	(No Documentation)
220-077	Lab Grades	10	10	(No Documentation)
220-078	FORTRAN Qubic	10	10	(No Documentation)
220-079	BASIC Qubic	10	10	(No Documentation)
220-080	PRINTX	10	10	
220-081	BASIC Curvefit	10	10	(No Documentation)
220-082	FORTRAN Macro Generator	15	25	
220-084	Gas Economy	10	10	
220-085	Cassette/Mag Tape Reversible Read Subroutines	10	15	
220-086	Battleships	10	10	(No Documentation)
220-087	Star Trek	20	40	
220-088	RTOS Snapshot	10	10	
220-089	TASK-S	10	15	
220-090	BASIC Print Utilities	10	10	
220-091	Text Processing-NOVA (Strip)	10	10	
220-092	Foreign Exchange Conversion	10	10	(No Documentation)
220-093	Auto Caller Driver Software	10	10	
220-095	UT200 Simulator	10	15	
220-096	NIM	10	10	(No Documentation)
220-097	MUMPS	10	10	(No Documentaiton)
220-100	RDOS/FORTRAN IV I/O	10	10	(No Documentation)
220-101	Character Processing	10	10	
220-102	RTOS/RDOS System Task	10	10	
220-103	DTECT	10	10	
220-104	Calendar	10	10	(No Documentation)
220-105	F1130 Program Converter	10	10	
220-106	Disassembler	10	15	
220-107	MCA Binary Loader	10	10	
220-108	File Compare	10	10	
220-109	NOVA Assembly Interpreter	10	10	
220-110	FORTRAN Random No. Generator	10	10	
220-111	Calculator	10	10	
220-113	D/A Converter Plotting Sub.	20	40	
220-115	BASIC Editor	10	10	
220-117	RDOS Binary Loader	10	10	
220-119	PREP	10	10	

Program No.	Name	Magnetic Tape (M)	Paper Tape (P)	
220-119	PREP	\$10	\$10	
220-120	Co-Pilot	10	15	
220-123	FORTRAN Post-Processor	10	10	(No Documentation)
220-124	Biorhythm	10	10	(No Documentation)
220-125	SOS Extended BASIC with RTC and Digital I/O	10	10	
220-126	Analysis of Variance Package	10	15	(No Documentation)
220-128	Stanford Business School Package	25	45	
220-129	NOVA Radio Music Interpreter	10	10	
220-131	Foreground CLI	10	10	(New Program)
220-133	Symbolic Alphanumeric Generator	10	10	(New Program)
220-134	Correlation Program	10	10	(New Program)
220-136	MYPRI	10	10	(New Program)
220-138	EBCDIC/ASCII Translator	10	10	(New Program)
220-139	PRNT Listing Routine	10	10	(New Program)
220-140	ALGOL Edit and Dump	10	10	(New Program)
220-141	BASIC Line Editor	10	10	(New Program)
220-145	FORTRAN IV Get Channel Routine	10	10	(New Program)
220-146	UNICO	10	10	(New Program)
220-147	IBMCO	10	10	(New Program)
220-148	ALGOL Date Routines	10	10	(New Program)
220-149	CLEAN	10	10	(New Program)
220-150	MAPCK	10	10	(New Program)
220-151	EIAAS	10	10	(New Program)
220-152	FORTRAN Error Message Interceptor	10	10	(New Program)
220-153	Directory Listing Program	10	10	(New Program)
220-154	L. M. U. Library	25	50	(New Program)
220-155	BRTIOS	10	10	(No Documentation)
220-156	POLAR	10	10	(New Program)
220-157	PAGE	10	10	(New Program)
220-158	DOTS	10	10	(New Program)
220-159	Program Space Memory Sizes	10	10	(New Program)
220-161	CRT to Disc Data Formatter	10	10	(New Program)
220-162	Function Diagram Printing Program	10	10	(New Program)
220-163	FORTRAN IV COM. CM Read	10	10	(New Program)
220-164	ASCII Fast Integer Tangent	10	10	(New Program)
220-165	EX Utility	10	10	(New Program)
220-184	Polyfit	10	10	(New Program)
220-185	Expfit	10	10	(New Program)
220-188	C/13 (A Chess Game)	10	15	(New Program)
220-199	CHAT Attribute Changing Program	10	10	(New Program)
220-200	Compile	10	10	(New Program)
220-201	Diskboot	10	10	(New Program)

Program No.	Name	Magnetic Tape (M)	Paper Tape (P)	
220-202	SBRTB	\$10	\$10	(New Program)
220-203	Tape Copy	10	10	(New Program)
220-204	MT Util.	10	15	(New Program)
220-205	RRAND Generator	10	10	(New Program)
220-206	GANDER	15	30	(New Program)
220-209	ASM16 Cross Assembler	10	15	(New Program)
220-210	ECLIPSE FORTRAN 5 Symbolic Debugger	10	15	(New Program)
220-211	Driver for LC50 R/P	10	15	(New Program)
220-214	LISP 1.5 Interpreter	15	15	(New Program)
220-215	BASIC Accounting	10	10	(New Program)
220-216	Biorhythm (ALGOL) Chart	10	10	(New Program)
220-217	ALEX. AL	10	10	(New Program)
220-218	ALGLIB	10	10	(New Program)
220-219	ALGPROC	10	10	(New Program)
220-220	SCRIBE	10	10	(New Program)
220-221	TANKA	10	10	(New Program)
220-222	ALFFT. LB	10	10	(New Program)
220-223	ALPLOT. LB	10	10	(New Program)

LIBRARY SETS

		Magnetic Tape (M)
220-900	Complete Library (Includes all current programs)	\$450
220-901	Games Volume I (Includes 27, 58, 59, 62, 64, 65, 66, 67, 72, 78, 79, 86)	\$ 80
220-902	Games Volume II (Includes 87, 96, 124, 129, 188)	\$ 45
220-903	Education Pack (Includes 46, 76, 77, 120)	\$ 25
220-904	Instrumentation/Control Pack (Includes 15, 19, 24, 37, 113, 125)	\$ 50
220-905	ALGOL Pack (Includes 49, 119, 123, 140, 148, 161, 184, 185, 216, 217, 218, 219, 220, 221, 222, 223)	\$100

MINIMUM ORDER - \$50.00







DATA GENERAL CORPORATION—WESTBORO, MASS. 01581 (617) 366-8911

INSTALLATION MEMBERSHIP FORM

Date _____

Name _____ Position _____

Company, Organization or School _____

Address _____

City _____

Telephone: Area Code _____ No. _____ Ext. _____

1. Account Category: OEM End User System House Government

2.

Hardware	Qty. Installed	Qty. On Order
C/330, C/300		
S/230, S/200		
S/130		
CS/40		
N3/D		
Other NOVA/3's		
microNOVA		
Other NOVAS		

3. Software: AOS DOS SOS RDOS RTOS Other Specify _____

4. Languages: Algol Assembler Basic Cobol Fortran RPG Other

5. Mode of Operation: Batch (Central) Batch (Via RJE) On-Line Interactive

6. Communications: RSTCP HASP RJE80 SAM CAM 4025 Other

Specify _____

INDUSTRY TYPE

7. 01 Agriculture/Mining Construction 09 Electrical Machinery Mfg.
 02 Transportation 10 Non-Electrical Machinery Mfg.
 03 Retail/Wholesale Trade 11 Other Manufacturing
 04 Finance/Banks/Insurance 12 Process Chemical/Petroleum Mfg.
 05 Medical/Hospital 13 Computer DP Services
 06 Education 14 Utilities
 07 Government 15 Printing/Publishing
 08 Services

8. Application Description _____

9. From whom was your machine(s) purchased?
Data General Corp. Other Specify _____

10. Are you interested in joining a special interest or regional Data General Users Group in your area? _____



Data General Corporation, Westboro, Massachusetts 01581, (617) 366-8911



Data General Corp. 046-008