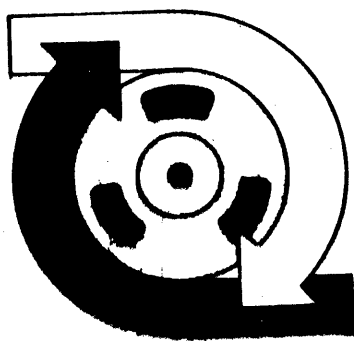


# Program Library Catalog



**INTERCHANGE**  
The Interdata Users Group

PROGRAM LIBRARY CATALOG INDEX

	<u>PAGE</u>
IUG-101: CLUE (DEBUG WITH TRACE)	1
102: SUPER-CLUB	1
103: TIC-TAC-TOE	1
104: MICA II IN-CORE ASSEMBLER	2
105: 360/20 SIMULATOR FOR MODEL 4	2
106: MODEL 3 ASSEMBLER	2
107: MODEL 3 SIMULATOR	3
108: TAPE LABEL MAKER	3
109: PAPER TAPE DUPLICATOR	3
110: DIAMOND DEMO	4
111: MAG-TAPE DRIVERS	4
112: UNIF-RANDOM NUMBER GENERATOR	5
113: IDS DISC UTILITY	5
114: TIDE W/MODIFY	6
115: TTY SUBROUTINES	6
116: FORTRAN EDIT UTILITY	6
117: A/D-READ	7
118: SUPER 50-SEQUENCE FOR MODEL 3	7
119: BOSS/3B (BASIC OPERATING SYSTEM)	8
120: CLOCK AND CALENDAR	9
121: LOAD MULTIPLE/STORE MULTIPLE TRAP	9
122: SIM-TRAP ROUTINES FOR LM, STM, PUSH, POP, AND TEXT EDITING	9
123: READ-WRITE-FUNCTIONS	10
124: SINK/SOURCE CONTROL	11
125: EDITOR/ASSEMBLER COMBO	11
126: MODEL 5 ASSEMBLER	12
127: 80-BYTE RECORD COPIER	12
128: MOD 1 FORTRAN COMPILER	13
129: MOD 1 FORTRAN RUN-TIME LIBRARY	13
130: OS CARD TO TAPE UTILITY	13
131: ASSEMBLER FOR MOD 1 W/64-BYTE ROM	14
132: TEXT EDITOR FOR MOD 1 W/64-BYTE ROM	14
133: LINEAR REGRESSION USING INTERACTIVE FORTRAN	15
134: GENERAL LOADER FOR WIRE WRAP TTY	15
135: RELOCATING LOADER FOR WIRE-WRAP TTY	15
136: TEXT EDITOR FOR WIRE WRAP-TTY	16
137: BOSS/4B FOR 16KB COMPILATIONS	16
138: BOSS/4B SYSTEM GENERATION SOURCE	16
139: MODEL 1 INTERTAPE PROGRAMMING SYSTEM (IPSI)	17
140: INTEGRATION USING TRAPAZOIDAL RULE	17
141: STAND-ALONE INTERTAPE DRIVER	17
142: OS MODEL 1 ASSEMBLER	18
143: MEDIA TO MEDIA PROGRAM	18

IUG-144:	OS TEXT EDITOR W/IMPROVED OUTPUT	18
145:	SOURCE DECK SYSTEM	19
146:	MODEL 1 MULTIPLY/DIVIDE SUBROUTINES	19
147:	PLOT4 - POOR MAN'S PLOTTER	19
148:	MAG TAPE UPDATE/MAINTENANCE PROGRAM	20
149:	RANDM - RANDOM NUMBER FUNCTION	20
150:	RANDU - RANDOM NUMBER GENERATOR	21
151:	VAPOR PRESSURE - TEMPERATURE TABLE	21
152:	SIMULATANEOUS EQUATIONS SOLVER	22
153:	MAG TAPE GENERAL LOADER	22
154:	BOSS/4A INTERTAPE DRIVER	22
155:	INTERDATA MODEL 70 CROSS ASSEMBLER FOR USE ON IBM 360/67 CP/CMS	23
156:	TELETYPE TEST PROGRAM (TELTYP)	23
157:	STORAGE OSCILLOSCOPE TEST (STOSCO)	23
158:	CORE I/O SIMULATOR DRIVER (BOSS/4A)	24
159:	FAST FORMAT PUNCHER FOR INTERTAPE	24
160:	FAST FORMAT LOADER FOR INTERTAPE	24
161:	FILER INDEX LISTING PROGRAM	25
162:	BOSS CORE DRIVER	26
163:	INTERTAPE TEST PROGRAM FOR MODEL 4	26
164:	INTER-SIG 707	26
165:	2-BYTE INTEGER FORTRAN IV COMPILER AND RUN-TIME LIBRARY	27
166:	BULLSEYE - THE ARTILLERY GAME	27
167:	YR - 20TH CENTURY DAY-OF-THE-WEEK GENERATOR	27
168:	OSOBAN - OS OBJECT TAPE ANALYZER	28
169:	DISK FILE SYSTEM	28
170:	ORI EDITOR	28
171:	INTEGER SQUARE ROOT SUBROUTINE FOR MODEL 5 AND NEW SERIES	29
172:	M10 TAPE GENERATOR (PAPER TAPE DUPLICATOR IN M10 FORMAT)	29
173:	CHARACTER TIME	29
174:	BOSS 4A BOOTTAPE CREATOR	30
175:	OS INT 4TRAN RW EXPANSION	30
176:	OS BLACK JACK	30
177:	YAHTZEE - A GAME PROGRAM IN BASIC	31
178:	GOLF IN BASIC	31
179:	ONE-ARM BANDIT GAME IN BASIC	31
180:	DATAPLOT 70	31
181:	OS DELOADER	32
182:	MAGTAPE DUMP	32
183:	MAGLDR	32
184:	BOSS/4D	33
185:	EBCDAS (SUBROUTINE TO CONVERT AN EBCDIC CHARACTER INTO ITS ASCII EQUIVALENT)	33
186:	GPM - GENERAL PURPOSE MACRO ASSEMBLER	33

IUG-187:	PLOT 10 - ACCESS LEVEL SOFTWARE AND GRAPHICS DEMO	34
188:	FAST FOURIER TRANSFORMS ON INTERDATA PROCESSORS	34
189:	SELECTOR CHANNEL READ/WRITE PACKAGE	34
190:	SYMBOL - TEXT-PRINTING USING INTERDATA 70 AND TEKTRONIX 4010	35
191:	SREAD - FORTRAN-CALLABLE SUBROUTINE	35
192:	TIMR86	35
193:	TIMR75	35
194:	ANALAD	36
195:	CLEAR	36
196:	COMAND	36
197:	DFIX	36
198:	EOFIND	36
199:	EOF85	36
200:	EXISER	37
201:	KHAR FUNCTION ROUTINE	37
202:	MOVEB ROUTINE	37
203:	MOVEIO	37
204:	MASKA ROUTINE	37
205:	OVLAY	38
206:	PROC IN INTERDATA FORTRAN	38
207:	PSWMOD IN ASSEMBLY LANGUAGE	38
208:	SHIFT	38
209:	SFTENT	38
210:	SIGATN	38
211:	SIGFRE	39
212:	SOURCE PROGRAM LIBRARY MAINTENANCE PROGRAM	39
213:	GAMES1	39
214:	MAG TAPE DEADSTART LOADER	40
215:	CROSS REFERENCE ASSEMBLER	40
216:	ARTILLERY GAME IN BASIC	40
217:	GOOK 1 & GOOK 2 IN BASIC	40
218:	GUESS WHAT? IN BASIC	40
219:	INTERDATA-IBM 360/370 CROSS ASSEMBLER	41
220:	FORTRAN CROSS ASSEMBLER	41
221:	OS XRFMT	42
222:	23 MATCHES GAME IN BASIC	42
223:	BLACK JACK GAME IN BASIC	42
224:	CHARACTER MANIPULATION & COMPARISON ROUTINES	42
225:	URAND	42
226:	DOS DISCTAN	42
227:	LINE FREQUENCY CLOCK DRIVER	42
228:	DOS BOOTSTRAP LOADER (FOR FIXED-HEAD DISC)	42
229:	RTOS DISCTAN	42
230:	LUPA/GETPA	43
231:	DIABLO DISK DRIVER FOR MODEL 4 AND BOSS/4B	43

IUG-232:	"UTILIZING A FIXED HEAD DISC UNDER DOS"	43
233:	BASIK - 132 CHARACTER BASIC (9 ZONES)	43
234:	LESSON IN BASIC - NUMBER ONE	43
235:	MEMORY DRIVER	43

\*NOTE - IUG-236 thru IUG-285 are written  
in Dartmouth Time Sharing System  
BASIC and are available in Listing  
form only.\*

236:	ANNUITY	44
237:	CALENDAR	44
238:	POPULA	44
239:	FORECAST (FCST)	44
240:	FOURIER	44
241:	RESPONSE	45
242:	LESSEE	45
243:	LESSIM	45
244:	LESSOR	45
245:	MODEL	45
246:	OPTION	46
247:	PROJECT	46
248:	SPREAD	46
249:	REPORT	46
250:	RISK	46
251:	GRAPH	47
252:	PLOT-IT	47
253:	PLOTSKI	47
254:	LEMT	47
255:	TEK10	47
256:	COMMON	47
257:	MAKE-BUY	48
258:	PROFITS	48
259:	CPM	48
260:	LAYOUT	48
261:	DIFEQ	48
262:	INTERP	48
263:	INTGRT	49
264:	QUADROOT	49
265:	RUNGE-2	49
266:	SIMPLEX	49
267:	ANOVAR	49
268:	LINFIT	49
269:	TEXTSUB	50
270:	SIMPREG	50
271:	STEPREG	50
272:	RUNOFF	50
273:	STENO	50
274:	STENOADD	50
275:	TEXTSUB	51

IUG-276:	VAL	51
277:	DRIVE	51
278:	PAYROLL SYSTEM	51
279:	GROUP	51
280:	SAMPSTAT	51
281:	STATSUB	52
282:	FLOW	52
283:	COMMENT	52
284:	MAILSORT	52
285:	MAILLIST	52
286:	DSCGEN - GENERAL LOADER FOR 2.5MB DISC	52
287:	OS AIDS/RTOS	53
288:	TAPTRN (TAPE TRANSLATOR)	53
289:	VIRTUAL DATA MEMORY	53
290:	INTERACTIVE FORTRAN W/TRAP	53
291:	INTERACTIVE FORTRAN/HARDWARE FLOATING POINT	53
292:	INTERACTIVE FORTRAN/SOFTWARE FLOATING POINT	54
293:	OS INTERACTIVE FORTRAN	54
294:	DOUBLE PRECISION INTEGER DIVISION (DBLDVD)	54
295:	DOUBLE PRECISION INTEGER CONVERSION ROUTINE (DBLINT)	54
296:	DOUBLE PRECISION INTEGER MULTIPLICATION (DBLMLT)	54
297:	INTERVAL TIMING (INTTIM)	54
298:	STAR TREK II	55
299:	STAR TREK III	55
300:	CLUB-DOS	55
301:	OS CLUE 70 (OS CLUB WITH TRACE)	55
302:	IFFUNC/BOSS 4B SCOPE DRIVER	55
303:	VARLIST	56
304:	XQDOS	56
305:	LOADER FORMAT ANALYZER	56
306:	GAMES 2	56
307:	INTEL 8008 & 8080	57
308:	COMPAR	57
309:	DSKCMP	57
310:	MAPSRT	57
311:	OLINTE	58
312:	SCOPYS	58
313:	SAVMEM	58
314:	UPDATO	58
315:	GTCAL	58
316:	DOSCMD (ICMD)	59
317:	DOS FOR MOD 4	59
318:	GEPAC30 PATCH MODULE TO INTERDATA CROSS REFERENCE ASSEMBLER	59
319:	SNAP2	59
320:	SSP - SCIENTIFIC SUBROUTINE PACKAGE	60

IUG-321:	INTERACTIVE GENERAL LOADER	60
322:	INTERCHANGE GAMES PACKAGE	61
323:	MOTOROLA M6800/INTERDATA CROSS ASSEMBLER	61
324:	PRINCIPLE, INTEREST RATE PROGRAM	61
325:	AMORTIZATION PROGRAM	62
326:	MATRIX ALGEBRA, LINEAR EQUATIONS	62
327:	TREND ANALYSIS PROGRAM	62
328:	GRAPH ROUTINE (TEKTRONIX 4010-1)	62
329:	SINE AND COSINE WAVES 1-720 DEGREES (TEKTRONIX 4010)	62
330:	PLTBIO (BIORHYTHM CHART PLOTTER)	62
331:	DISCUPT UTILITY (DISC PATCH)	63
332:	MULTI-USER INTERACTIVE FORTRAN SYSTEM	63
333:	PUNCH	63
334:	SOURCE MAINTENANCE UTILITY TASK	64
335:	UT 200 EMULATOR FOR DEDICATED 7/16	64
336:	UT 200 EMULATOR FOR 7/32	64
337:	DCT-1000 MULTIPLEXOR EMULATOR FOR UNIVAC 1100	64
338:	GTCALRO2 - FORTRAN VERSION OF CAL	64
339:	EXPANDED PRECISION ARITHMETIC PACKAGE - EPAP	64
340:	OS/32MT UNIVERSAL CLOCK MODULE & UNIVERSAL LOGIC INTERFACE DRIVER	64
341:	70X LOADER FORMAT ANALYZER	65
342:	DISCREL	65
343:	DIRECT	65
344:	PSEUDO-RANDOM NUMBER GENERATOR SUBROUTINE PACKAGE	65
345:	OS/32 ASCEBC	65
346:	EBCASC	65
347:	1130 COMMERCIAL SUBROUTINE PACKAGE	66
348:	CANONICAL CORRELATION ANALYSIS - CANONA	66
349:	NSTAT - NONPARAMETRIC STATISTICAL ANALYSIS & CROSS CLASSIFICATION	66
350:	7-TRACK MAGNETIC TAPE DRIVER	66
351:	DISCUPD UTILITY	67
352:	OS/16MT-2 SYSTEM TABLE BUILDER II	67
353:	FIND	67
354:	FETCH	67
355:	SCAN	67
356:	TIC-TAC-TOE	68
358:	CRIBBAGE	68
359:	SNAPSHOT	68
360:	MULTI-USER EXECUTIVE OPERATING SYSTEM FOR 16-BIT CPU	68
361:	FIXED AND FLOATING POINT MATH PACKAGE	69
362:	TIDE (16-BIT TEXT EDITOR)	69
363:	RTOS - REAL TIME OPERATING SYSTEM - 16-BIT MODEL 70/80	69

IUG-364:	RTEX - REAL TIME TELECOMMUNICATIONS EXECUTIVE	69
365:	BASIC INTERPRETER	70
366:	FLOXOS - REAL TIME BASIC WITH FLOPPY DOS	70
367:	OBJDUMP - OBJECT DUMP	70
368:	CAL-360/370 CROSS ASSEMBLER	70
369:	MOVESTR	71
370:	BFP - BASIC FORTRAN PLOTTER	71
371:	FBCOPY - FIXED-BLOCK AND UNBLOCK COPY	71
372:	OS/32MT RUN ANALYZER	72
373:	SIMULATION PROGRAMS IN BASIC LEVEL I	72
374:	SFS - SPOON FEEDER SYSTEM	72
375:	SORT	73
376:	CAL 360/370 CROSS ASSEMBLER	73
377:	INPUT	73
378:	SNAP	73
379:	DATA DIRECTOR EDITOR	74
380:	BFP (BASIC FORTRAN PLOTTER)	74
381:	SOLO OPERATING SYSTEM INCLUDING SEQUENTIAL AND CONCURRENT PASCAL INTERPRETERS	75
382:	COBOL SQUEEZER	75
	Request Form	76
	Submittal Form	77, 78
	Program Review Form	79, 80



INTERCHANGE PROGRAM LIBRARY CATALOG

ABSTRACTS

IUG-101: CLUE (DEBUG WITH TRACE)

System Requirements:

Mod 3, TTY, CLUB, 03-013R00,  
Rel/Gen Loader

Occupies: X'522' Bytes Above CLUB

Language: Interdata Basic Assembler

Available material:

Relocatable Object Tape  
Assembly listing, description

CLUE is an extension of the Interdata Hexadecimal Debug Program, 03-013. The operation of CLUE is identical to that of CLUB with Disassembly and Output, as described in the CLUB manual, Publication Number 03-002R03A12, except for the addition of directives for selective program trace, sequential program input, and extension of the breakpoint features. Linkage to the 03-013R00 version of CLUB is required at load time.

IUG-102 : SUPER-CLUB

System Requirements:

Mod 3, TTY  
50-Sequence Loader

Language: Interdata Basic Assembler

Available Material:

Bootstrap Object Tape, Description

SUPER-CLUB is a combination of the Interdata General Loader, 06-025R01, as described in the Loader Description, 29-231, and the Hex-Debug (CLUB) program, 03-013R02, as described in the CLUB Manual, 29-235. An asterisk (\*) directive, added to Club, allows the setting of the loader bias and transfer to the loader. The loader returns to Super-Club following each load. Continuous linkage loading is provided by repetitive asterisk (\*) directives. Any interrupts which occur are trapped, and a message is typed. The program is available in bootstrap form which is self-loading into the topmost 4K bytes of memory.

IUG-103: TIC-TAC-TOE

System Requirements:

Mod 3, TTY,  
Rel/Gen Loader

Occupies: X'A50' Bytes

Language: Interdata Basic Assembler

Available material:

Relocatable object tape,  
Assembly listing, description

This demonstration program plays Tic-Tac-Toe on from 1-16 teletypes having addresses X'00' to X'0F'. This game is played interactively at each teletype, with the print-out showing the status of the game and a goading message after each play. The program involves four levels of strategy, ranging from Random to Unbeatable. As the game is played, the computer becomes more and more difficult to beat each time it loses.

IUG-104 MICA II IN-CORE ASSEMBLER

System Requirements:	Language: INTERDATA Basic Assembler
Mod 4, TTY	Available Material:
GEN Loader	Relocatable Object Tape(s)
Occupies:	Assembly Listing(s)
2KB	Description

This small interactive Assembler for the Model 4 occupies only 2K bytes of memory. Instructions and data are typed by the user on the console Teletype. The instructions and data are assembled directly into core without the necessity of punching a paper tape which is to be loaded into core. No relocation facilities or symbolic addressing is available, although addressing relative to the current location is permitted. The program tape contains six sub-programs requiring the General Loader for linkage loading.

IUG-105: 360/20 SIMULATOR FOR MODEL 4

System Requirements:	Language: INTERDATA Basic Assembler
Mod 4, 8KB, TTY	Available Material:
HS Opt for Mag Tape I/O	Relocatable Object Tape
REL/GEN Loader	Assembly Listing
Occupies:	Description
X'F84' Bytes	

Simulated execution of 360/20 object code is performed by this program. The 360/20 program occupies a portion of memory along with the core resident simulator which requires 4K. A Model 4 with at least 8K bytes of memory, is required. All 360/20 instructions, including I/O and decimal arithmetic (with multiply and divide) can be executed directly with no change to the 360 program. The simulator executes 360/20 instructions at about the same speed as the Model 20 itself.

IUG-106: MODEL 3 ASSEMBLER

System Requirements:	Language: BASIC for GE Time-Share Systems
GE Time-Share System	Available Material:
	Source Listing Only

This program is an interactive assembler written in Basic, to run on the GE Time-Share System. The assembler consists of several modules for file handling, Pass 1 and 2, listing generation, and tape punching. Use of the program requires some knowledge of the file handling and control language of the time-share system. This program is available only as a Basic listing, with no supporting documentation. The listing may be of interest to prospective users of the GE system, or as an example of an assembler written in Basic.

IUG-107: MODEL 3 SIMULATOR  
System Requirements:  
MIT 360 Time-Share System

Language: 360 BAL for 360/67 Time-Share System  
Available Material:  
Source Listing

This program, which runs on the MIT 360 time share system, simulates the execution of INTERDATA Model 3 object code. It provides at a time sharing terminal, interactive features for program creation, execution, tracing, and breakpointing. The program is a complete simulation of all Model 3 instructions with special provisions for handling I/O instructions at the terminal. This program is available as a listing of the source program, which is written in 360 Assembly language.

IUG-108: TAPE LABEL MAKER

System Requirements:  
Mod 3, TTY  
ABS/REL/GEN Loader  
Occupies:  
X'0080' to X'03D2'

Language: INTERDATA Basic Assembler  
Available Material:  
Absolute Object Tape  
Assembly Listing  
Description

The Tape Label Maker is used to make a readable label on a punch paper tape. A line of characters can be entered from the teletype keyboard, edited as needed and punched on the tape in a readable form.

IUG-109: PAPER TAPE DUPLICATOR

System Requirements:  
Mod 3, HSPTR, HSPTP  
ABS/REL/GEN Loader  
Occupies:  
X'80' to X'1C0' using  
remaining core for tape  
image

Language: INTERDATA Basic Assembler  
Available Material:  
Absolute Object Tape  
Assembly Listing  
Description

This program allows the duplication and verification of paper tapes using the high speed paper tape reader and punch. A master tape is read and stored in its entirety in core memory. The master tape is then re-read for verification. The tape is duplicated by punching the data from core memory to paper tape, with appropriate leader and trailer. A number of copies can be made as specified by a count in display panel switches. A verify mode then allows checking of all copies punched. This program requires less than 500 bytes, but 16K memory or more is required for duplication of large programs.

IUG-110: DIAMOND DEMO

System Requirements:

Mod 3, X-Y Scope  
REL/GEN Loader

Occupies:

X'F0' Bytes

Language: INTERDATA Basic Assembler

Available Material:

Relocatable Object Tape  
Assembly Listing  
Description

This program is designed to provide a demonstration package for a Model 3 with an oscilloscope. The program generates two triangular wave shapes that are 90° out of phase. One wave shape is fed to the X axis, the other to the Y axis of the scope. The resultant diamond pattern appears on the scope face. Parameters in the program control the picture intensity.

IUG-111: MAG-TAPE DRIVERS

System Requirements:

Mod 4, HS Opt, Mag Tape  
GEN Loader

Language: INTERDATA Basic Assembler

Available Material:

Relocatable Object Tape  
Assembly Listing  
Description

The Mag Tape Driver Routines provide 8 subroutines for handling normal data transfer operations with IBM compatible 7-track or 9-track magnetic tape:

Write A Record  
Read A Record  
Backspace n Records  
Backspace n Files  
Advance n Records  
Advance n Files  
Write End Of File  
Rewind

The Read routine includes error detection and recovery procedures. Each operation is started with a Branch and Link subroutine call to specify ENTRY/EXTRN global symbols. Status of the I/O operation is returned to the calling program.

IUG-112: UNIF-RANDOM NUMBER GENERATOR

System Requirements:	Language: INTERDATA Basic Assembler
Mod 3, HS Opt	Available Material:
(MH Instruction)	Relocatable Object Tape
REL/GEN Loader	Assembly Listing
Occupies:	Description
X'B0' Bytes	

UNIF generates ten uniformly distributed random numbers between X'0000' and X'FFFF'. The calling sequence is:

```
BAL 15,UNIF
DC A(RSLT1)
DC A(RSLT2)
```

RSLT1 is any full word odd integer between X'0000', X'0001' and X'01FF', X'FFFF' which is initially defined by the programmer and is subsequently redefined by UNIF. RSLT2 is the string of uniform halfword integers. The uniformly distributed numbers are generated as follows:

$$RSLT1 = \text{modulo } \{ (RSLT1) * (3125) \}$$
$$RSLT2 = \{ RSLT1 / 2^{16} \}$$

For ten iterations where modulo arithmetic is  $2^{26}$  and {A} means the largest integer is not greater than A. All general registers used are saved and restored upon return.

IUG-113: IDS DISC UTILITY

System Requirements:	Language: INTERDATA Basic Assembler
Mod 3, 16KB, TTY, Selch,	Available Material:
IDS Disc Mod 7032	Absolute Object Tape (CLUB "0" Dump)
HEX-Debug, 03-013R00	CLUB Disassembly Listing w/comments
at X'2300'	
ABS/REL/GEN Loader	
Occupies: X'2D00' to X'2F5C'	

This program provides a set of Read/Write/Compare utilities for the IDS Disc, Model 7032. This program can write from selected areas of core in 200 halfword increments, read into a buffer and compare. Errors cause a type-out showing address and contents. The program can also read from a selected track and sector into core. Also, it is possible to write, read or compare through the disc, without incrementing core locations. The program, which is absolute, assumes CLUB is at location X'2300', and utilizes its illegal instruction trap.

IUG-114: TIDE W/MODIFY

System Requirements:

Mod 3, TTY  
REL/GEN Loader

Occupies: X'BC0' Bytes  
plus adjustable 1KB  
text buffer

Language: INTERDATA Basic Assembler

Available Material:

Relocatable Object Tape  
Assembly Listing  
Description of M or MV Commands

This program is the standard version of TIDE, 06-014R00, as described in TIDE manual 06-014A15, with the addition of two new commands: M for Modify or MV for Modify-Verify. This command allows modification of a character string within the open line. Only enough characters need to be typed to specify the particular string to be modified. Characters may be changed, added or inserted, or deleted.

IUG-115: TTY SUBROUTINES

System Requirements:

Mod 4, HS Opt, TTY  
GEN Loader

Occupies:  
X'2AA' Bytes

Language: INTERDATA Basic Assembler

Available Material:

Relocatable Object Tape  
Assembly Listing  
Description

This program consists of a group of utility subroutines which will handle most teletype functions. They include standard input, output message as well as special character commands. In the latter case, upon recognition of the special character, control is transferred to a core location which contains a pointer to a service routine (or other user program). Each function is accomplished by the user's programmed calling sequences with BAL's and global symbols specified in the description.

IUG-116: FORTRAN EDIT UTILITY

System Requirements:

Mod 4, HS Opt, Fltg. Pt.,  
16KB

BOSS, FORTRAN IV Compiler  
& Routine Library  
CR/MT/DRUM/DISC

Language: FORTRAN IV

Available Material:

Compiled Listing

The Fortran Edit Utility is useful for updating a source program file on a magnetic tape or equivalent device. The program is useful as an example of character-manipulation in the Fortran IV language. The compiled program uses BOSS for logical I/O, and is therefore device independent. The functions of this program are:

1. Place source statements (cards) onto magnetic tape with sequence number.
2. Modify existing source on mag tape and inset update statements by sequence number.
3. Re-sequence existing cards on magnetic tape.
4. List any card image file.

This program is provided in listing form only, which includes many comments. The program includes less than 50 statements and must be compiled to be used.

IUG-117: A/D-READ

System Requirements:  
 Mod 3, A/D Converter  
 REL/GEN Loader

Occupies:  
 X'48' Bytes

Language: INTERDATA Basic Assembler  
 Available Material:  
 Relocatable Object Tape  
 Assembly Listing  
 Description

AD-READ is a subroutine for use in reading a preset number of A/D converter channels sequentially. Each channel, in the case of an 8-bit converter, uses one byte and, in the case of 10-bit or 12-bit converters, uses two bytes. The Calling Sequence requires that the starting address of the data buffer for the particular pass of the A/D Converter is loaded into Register 6, the last buffer address is loaded into Register 8, and the increment into Register 7. All other registers used are restored. The address which contains the first channel number to be queried, is a constant following the branch and link instruction.

IUG-118: SUPER 50-SEQUENCE FOR MODEL 3

System Requirements:  
 Mod 3 ONLY

Language: INTERDATA Basic Assembler  
 Available Material:  
 Listing Below

This new 50 sequence for the Model 3 does not require bootstrap tapes to be loaded on the first character:

```

0050 C830 LOAD LHI 3,-X'4F'
      FFB1
0054 OBCC SHR 12,12
0056 D3A0 LB 10,BINDV
      0078
005A DEAO OC 10,BINDV+1
      0079
005E 01EE BALR 14,14
0060 9DAD SSR 10,13
0062 02FE BTCR 15,14
0064 9BAD RDR 10,13
0066 OACD AHR 12,13
0068 033E BFCR 3,14
006A D2D3 STB 13,X'80'+X'4F'(3)
      00CF
006E CA30 AHI 3,1
      0001
0072 032E BFCR 2,14
0074 4300 BFC 0,X'80'
      0080
0078 0394 BINDV DC X'0394'
007A 0394 BOUTDV DC X'0298'
007C 0394 SINDV DC X'0394'
007E 0298 SOUTDV DC X'0298'
0080 END

```

This 50 Sequence is operable only on the Model 3, because of the nature in which Branch and Link (BALR) is decoded in the Model 3 firmware. The Model 3 BALR performs linkage prior to branching whereas the Model 4/5 branches prior to linkage.

IUG-119: BOSS/3B (BASIC OPERATING SYSTEM)

System Requirements:

Mod 3, TTY

Occupies:

X'D00' Bytes

Language: INTERDATA Basic Assembler

Available Material:

Bootstrap Object Tape

Assembly Listing

BOSS/3B is a Basic Operating System which runs on the INTERDATA Model 3, and requires a little more than 3KB of core (X'0D00'). BOSS/3B acts in every way like BOSS/4B and conforms with the Basic Operating System (BOSS) Manual No. B29-216. BOSS/3B uses the illegal instruction PSW to trap out the Supervisor Call (SVC), Store Multiple (STM), and Load Multiple (LM) instructions and perform them in software. The trap routines are written so that the programmer has full use of the instructions for his own use as well as BOSS's use.

Note: There is an error in the above tape.

Location 856 contains 7CC - this should be 7D0



IUG-120: CLOCK AND CALENDAR

System Requirements:

Mod 3, 60Hz Real Time  
Clock

Language: INTERDATA Basic Assembler

Available Material:  
Source Listing

This program is an example, in source listing form only, of calendar time tracking by use of a real-time 60Hz clock. With this subroutine, the user need only append an interrupt handling routine to obtain the 60 cycle count, and an initialization routine to preset the calendar date in six halfwords (secs, mins, hours, day, month, years) in conjunction with enabling interrupts on a precision 60Hz real-time clock.

IUC-121: LOAD MULTIPLE/STORE MULTIPLE TRAP

System Requirements:

Mod 3  
GEN Loader

Occupies:

X'020A' Bytes (Version 1)  
X'021C' Bytes (Version 2)  
X'006A' Bytes (STACK)

Language: INTERDATA Basic Assembler

Available Material:  
Relocatable Object Tape  
(w/2 versions + stack)  
3 Assembly Listings  
Description

The Load Multiple/Store Multiple Trap routine is a re-entrant trap routine that simulates execution of the LM/STM instructions. This program allows a user to run any program using Standard Model 4 instructions (except Autoload and Model 4 Options) on the Model 3. Illegal Instruction trap routines, stack overflow/ underflow routines, or interrupt handling routines using LM and STM, may be used with this program as specified in the description.

IUG-122: SIM-TRAP ROUTINES FOR LM, STM, PUSH, POP, AND TEXT EDITING

System Requirements:

Mod 3, TTY  
HEX-DEBUG, 03-013R00 at  
X'80'  
GEN Loader

Language: INTERDATA Basic Assembler

Available Material:  
Assembly Listings  
Description

SIM is the main program of a set of routines designed to allow the use of certain optional instructions on machines without their hardware implementation. These instructions are STM, LM, PUSH, POP, CLBR, CLB, MOVR, MOV, FNDR, and FND. The General Loader is required for ENTRY/EXTRN linkage. The set of routines is independent of all other executing programs in the machine since entry and exit are achieved via illegal instruction interrupts and PSW swapping.

## IUG-123: READ-WRITE-FUNCTIONS

### System Requirements:

Mod 4, HS Opt, Fltg. Pt.

16KB

HSPTR/P or Card Reader

Interactive Fortran

w/RWF, 03-007R00

REL Loader

Language: INTERDATA OS Assembler

Available Material:

Relocatable Object Tape

Assembly Listing

Description w/Example Printout

### Occupies:

X'3D8' Bytes

This program provides some READ-WRITE-FUNCTIONS for INTERDATA's Interactive Fortran with RWF, 03-007R00, such as:

- READ: Card Reader and High Speed Paper Tape  
\*Uses conversion codes for IBM-026 Keypunch Hollerith with changes available on listing of conversion codes for IBM-029 Keypunch Hollerith in Card Reader routine.
- WRITE: High Speed Paper Tape
- FUNCTIONS: Integer/Fraction, Random numbers, and Absolute value.

Expansion slots are provided to facilitate other user I/O device handlers and functions. Additional Interactive Fortran user statements implemented are:

READ X,A,B,C,... reads one statement from the Card Reader (X=4) or from the High Speed Paper Tape Reader (X=19<sub>10</sub>). Variables A,B,C... are set to the respective values separated by commas on the statement read.

WRITE X,A,B,C,... writes the values of the variables, expressions or character Strings A,B,C... to the High Speed Paper Tape Punch when X=19<sub>10</sub>.

WRITE X writes 200 null characters as leader or trailer with punch-off control to the High Speed Paper Tape Punch when X=20<sub>10</sub>.

FUNC (X,A), NUMBER initializes the Random Number Generator with NUMBER (e.g. 7305.1) (X=5) and FUNC (X,A) causes A to be set to a generated random number when X=5.

FUNC (X,A),B when X=10<sub>10</sub> causes the variable A to be set to the absolute value of the variable B.

FUNC (X,A,B),C when X=3 causes A to be set to the integer portion of the variable C and B is set to the fractional portion of the variable C.

IUG-124: SINK/SOURCE CONTROL

System Requirements:

Mod 4, HS Opt, Fltg. Pt.  
16KB  
HSPTR or Card Reader  
Interactive Fortran  
w/RWF, 03-007R00  
IUG-123, RWF's  
REL Loader

Language: INTERDATA OS Assembler

Available Material:

Relocatable Object Tape  
Assembly Listing  
Description w/Example Printout

Occupies:

X'B0' Bytes

This program, in conjunction with IUG-123 READ-WRITE-FUNCTION, provides Sink/Source control for Interactive Fortran programs. It allows user Interactive Fortran programs to be entered via the Card Reader or the High Speed Paper Tape Reader by means of the additional system command statement in Direct Mode, " ← RCRD." By patching a halfword in IUG-124 containing the device number, the " ← RCRD" statement reads from the HSPTR.

IUG-125: EDITOR/ASSEMBLER COMBO

System Requirements:

Mod 3, 16KB, TTY, HSPTR=X'03'

Occupies:

X'78' to X'28A6' using  
16KB for text buffer

Language: INTERDATA Basic Assembler

Available Material:

Bootstrap Object Tape  
Assembly Listings  
Description

This program modifies the standard INTERDATA Basic Assembler, 03-024, and original Editor (TIDE), 06-014, in three forms:

- (1) those which extend the editing powers of TIDE, including modifications which allow the user to choose between two different command input devices, the standard teletype and the ARDS or Advanced Remote Display Station, as well as two different punch devices, in this case, a standard teletype punch and a high-speed punch.
- (2) those which allow the user to move between the Editor and the Assembler
- (3) those which permit the user to specify list and punch devices for the assembler to operate on, as well as the format of output which the assembler generates.

In keeping with these changes, all previous TIDE commands have been preserved and the new commands have been implemented with identical format and control. In addition to the description provided, user information on this assembler is found in 29-230 and on the editor in 06-014A12.

IUG-126: MODEL 5 ASSEMBLER  
System Requirements:  
360/67 Time-Share System

Language: 360 BAL  
Available Material:

The Model 5 Assembler was contributed by Nicholas Negroponete of M.I.T. The assembler has all the features of the standard basic assembler with the restriction that floating-point constants and expressions cannot be used. The system is written to run with the CMS system on the 360/67, and its I/O structure is dependent on that system. Professor Negroponete has indicated that this program could be modified for other I/O systems, and he has offered to provide the program to interested users who will send him a mag tape reel on which he can write the program. Interested parties should contact the INTERCHANGE Secretary for further information on this program.

IUG-127: 80-BYTE RECORD COPIER  
System Requirements:  
Mod 4, 8KB, BOSS, TTY  
Occupies:  
X'194' Bytes

Language: INTERDATA OS Assembler  
Available Material:  
Relocatable Object Tape  
Assembly Listing

The Copier program is used to copy 80-byte source records from logical unit 1 to logical unit 2. Double buffering is employed to make use of the overlapping I/O capability of BOSS/4A. If a non-zero status is detected, it is printed on the console device and a PAUSE SVC is issued. On continuing, the operation for which the non-zero status was returned, is re-tried.

IUG-128: MOD1 FORTRAN COMPILER

System Requirements:

Mod 1, 4K, TTY  
Mod 1 GEN Loader

Occupies:  
3.5KB

Language: Mod 1 Assembler

Available Material:

Absolute Object Tape  
Mod 1 Assembly Listing  
Language Specification and  
User's Guide

The Model 1 Fortran System consists of IUG-128, a Basic Fortran Compiler and IUG-129, Mod 1 Fortran Run-Time Library routines designed to operate on a Model 1 with 4KB memory. A teletype is the only required device. The compiler is fully resident and requires only one pass across the source. It reads a source tape producing a listing, dictionary table, and user object tape. In 4KB, the compiler allows about 500 bytes for dictionary space.

IUG-129: MOD 1 FORTRAN RUN-TIME LIBRARY

System Requirements:

Mod 1, 4K, TTY  
Mod 1 GEN Loader

Occupies:  
3KB

Language: Mod 1 Assembler

Available Material:

Absolute Object Tape  
Mod 1 Assembly Listing  
Language Specification and  
User's Guide

The Model 1 FORTRAN System consists of IUG-128, a Basic FORTRAN Compiler and IUG-129, the Mod 1 FORTRAN Run-Time Library which is provided as a single, absolute, low core resident tape. The entire library is about 3KB, but the User object program may overlay upper sections (1K) of the Run-Time Library to increase user program size, i.e., if TRIG, LOGARITHMIC, or other functions are not required). Without overlays, 1KB of user program space is available on a 4KB system with the Run-Time Library.

IUG-130: OS CARD TO TAPE UTILITY

System Requirements:

Mod 4, 8KB, BOSS

Occupies:  
X'11D2' Bytes

Language: INTERDATA OS Assembler

Available Material:

Relocatable Object Tape  
Assembly Listing  
Description

This program, in conjunction with a resident operating system, converts source information from logical unit 1 to logical unit 2 in identical or contracted standard INTERDATA source format, such as from cards to paper tape or magnetic tape. Upon execution at its origin, message/response interaction with the operator at logical unit 00/05 provides self-explanatory direction in regard to operating procedures. Verification of the generated source on L'2 is accomplished by a second pass across the source on LUL. Listings can be obtained on LU3 of the LUL source during the first reproduction pass or of the LU2 source during the second verification pass. An optional sequence check for ascending sequence identifiers on each LUL source statement is provided on both passes. An optional "squeeze" of the LUL source is provided: (deletion of

sequence field, left justification of carriage return to last non-space character of each line, and reduction of multiple spaces to a single delimiting space). Reproducing large source programs to paper tape allows the creation of "multitapes", whereby every one-inch thicknesses of fan-fold paper tape, an INTERDATA OS Assembler language "PAUSE" statement is injected on the LU2 copy, and the operator obtains trailer, breaks the tape, obtains leader and continues.

IUG-131: ASSEMBLER FOR MOD 1 W/64-BYTE ROM

System Requirements:	Language: INTERDATA Mod 1 Assembler
Mod 1, 4K, TTY	Available Material:
Mod 1 GEN Loader	Absolute Object Tape
Occupies:	Mod 1 Assembly Listing
3KB	

This program is the Model 1 Assembler, 08-004, modified to allow the user with a 64-byte crushed core loader (ROM) to perform assemblies. The functions and specifications of this program are identical to those described in the Model One User's Manual, 29-215.

IUG-132: TEXT EDITOR FOR MOD1 W/64-BYTE ROM

System Requirements:	Language: INTERDATA Mod 1 Assembler
Mod 1, 4K, TTY	Available Material:
Mod 1 GEN Loader	Absolute Object Tape
Occupies:	Mod 1 Assembly Listing
2KB	

This program is the Model 1 Text Editor, 08-007, as modified to allow the user with a 64-byte crushed core loader (ROM) to edit text. The functions and specifications of the program are identical to those described for the standard editor in the Model One User's Manual, 29-215, with the following exceptions:

Special Text Editor Address

'0040'	Restart location, program will not initialize text buffer
'0044'	Starting location, program will initialize text buffer
'0048'	Defines first address of text buffer
'004A'	Defines last address of text buffer

IUG-133: LINEAR REGRESSION USING INTERACTIVE FORTRAN

System Requirements:	Language: Interactive Fortran
Mod 3, 16KB, TTY	Available Material:
and Interactive FORTRAN	Source Listing (as created by
With TRAP, 03-006R02	03-006R02)
or	Source Tape (as created by 03-006R02)
Mod 4, HS Opt, (Fltg. Pt),	Description
TTY	
and Interactive FORTRAN/	
MOD4 or RWF, 03-007 or	
03-011	

This program is designed to perform a regression analysis on two variables. It gives the standard error of estimate, a least square best fit line, and a table of residuals. (The source tape can be converted through 03 007R01 to be useful with OS Interactive Fortran, 03-033.) Note: the original Interactive Fortran's separated statements with ASCII spaces and the upcoming I.F. revisions have standardized inter-record paper tape gaps of 8 nulls.

IUG-134: GENERAL LOADER FOR WIRE WRAP TTY

System Requirements:	Language: INTERDATA OS Assembler
Mod 3, TTY	Available Material:
50 Sequence Loader	Bootstrap Object Tape
Occupies: X'600' bytes from	Assembly Listing
top-of-core	

This program is the INTERDATA General Loader, 06-025R01, as modified to be loaded and to load in a wire-wrap TTY environment. User description is available in the Loader Descriptions, Publication Number 29-231R01.

IUG-135: RELOCATING LOADER FOR WIRE-WRAP TTY

System Requirements:	Language: INTERDATA OS Assembler
Mod 3, TTY	Available Material:
50 Sequence Loader	Bootstrap Object Tape
Occupies:	OS Assembly Listing
X'400' Bytes from	
top-of-core	

This program is the INTERDATA Relocating Loader, 06-024R01, as modified to be loaded and to load in a wire wrap TTY environment. User description is available in the Loader Descriptions, Publication 29-231R01.

IUG-136: TEXT EDITOR FOR WIRE WRAP TTY

System Requirements:	Language: INTERDATA OS Assembler
Mod 3, TTY	Available Material:
REL/GEN Loader	Relocatable Object Tape
Occupies:	OS Assembly Listing
X'13FA' Bytes	

This program is the INTERDATA TIDE Editor, 03-026R01, as modified for I/O in a wire wrap TTY environment. User description is available in the Editor (TIDE) Program Manual, Publication No. 29-229R01.

IUG-137: BOSS/4B FOR 16KB COMPILATIONS

System Requirements:	Language: INTERDATA OS Assembler
Mod 4, 8KB, TTY	Available Material:
HS Opt for Mag Tape	Combined Bootstrap & Relocatable
(For use in FORTRAN IV	Object Tape
compilations:	OS Assembly Listing of modified
Mod 4, HS Opt,	BOSS/4B
Fltg. Pt., 16KB	Description
TTY, FORTRAN IV Compiler)	

This program is a modified version of BOSS/4B, 03-021R01, with the loader and magnetic tape driver routines removed. Its implicit purpose is to facilitate FORTRAN IV Compilations under BOSS/4B in 16KB environments with directions available in the IUG-137 description. User information on compilations under BOSS/4B are found in BOSS Program Manual, B29-216 and the BOSS FORTRAN IV User's Guide, 29-246. The program tape combines a bootstrapped Rel Loader followed by the modified BOSS/4B such that not only is IUG-137, in effect, self-loading but that the compiler can then be loaded by the same Rel Loader.

IUG-138: BOSS/4B SYSTEM GENERATION SOURCE

System Requirements:	Language: INTERDATA OS Assembler
OS Assembler	Available Material:
	Source Multitape (3 parts)
	Source Listing of Multitape
	Description

This program is the source form of BOSS/4B, 03-021R01, in multitape form to provide users with the ability to add or delete modules from BOSS/4B to conform to individual system requirements. The source tape consists of 3 sections with source statements squeezed (no sequence fields, left-justified terminal carriage return, and multiple spaces reduced to a single delimiting space). It contains no OPTION statement and must be assembled with the OS Assembler, 03-025, as each separate section ends with a "PAUSE" statement and TITLE and IF statements are used.



IUG-139: MODEL 1 INTERTAPE PROGRAMMING SYSTEM (IPSI)

System Requirements:	Language: Mod 1 Assembler
Mod 1, 4KB	Available Material:
SERIAL TTY	Absolute Object Tape
1 or 2 INTERTAPE Systems	Assembly Listing
Mod 1 General Loader	Description

Occupies: '800' - 'FFA'

The Intertape Programming System, (IPSI) for the INTERDATA Mod 1 processor is a cassette resident operating system. IPSI has two main functions. It allows the building of a system library on cassette which may contain Model 1 system programs, user-written programs, or user data files. Elements entered into the library may be labeled and called by name with keyboard commands. Secondly, IPSI provides cassette and teletype Input/Output routines which may be called by user programs. In addition, IPSI contains other cassette utility commands and debugging features.

IUG-140: INTEGRATION USING TRAPAZOIDAL RULE

System Requirements	Language: FORTRAN IV
Mod 4, High Speed Option,	Available Material:
Floating Point, 16KB	Description
BOSS/4B, FORTRAN IV Compiler,	Listing
Run-Time Library, FUNC.	

This program performs integration using Trapezoidal Rule on any polynomial of the form  $C(1) + C(2) X + \dots + C(I+1) X^I$  up to order 99 where the user specifies:

1. Order of the Polynomial = I
2. Co-efficients of the Polynomial = C(i)
3. Lower Limit, Upper Limit, Number of Intervals  
 $+1 = N$ ; where N may range from  $2 \leq N \leq 999$

The program and its associated Function subroutine (FUNC) are provided in listing form only.

IUG-141R01: STAND-ALONE INTERTAPE DRIVER

System Requirements:	Language: OS Assembler
Mod 4, INTERTAPE	Available Material:
Occupies:	Relocatable Object Tape
X'0222' Bytes	Assembly Listing
	Description

This stand-alone driver performs the following I/O operations with the INTERTAPE cassettes: READ, WRITE, REWIND, WRITE AN END-OF-FILE, SKIP-RECORD-FORWARD, SKIP-RECORD-REVERSE, SKIP-EOF-FORWARD, and SKIP-EOF-REVERSE. No High Speed Option or Selector Channel is required. Appropriate call sequences are as follows:

For READ/WRITE:	BAL R15,CASDVR	For COMMANDS:	BAL R15,CASDVR
	DC X'xyy'		DC X'xyy'
	DC O		DC O
	DC A(START)		
	DC A(END)		

where xx=function code and yv=physical device address of cassette drive. The R01 version removes an erroneous DC 0 from the R00 object tape and adjusts the address arithmetic used to allow blocks of data greater than 32KB to be READ/WRTTEN.

IUG-142: OS MODEL 1 ASSEMBLER

Author: Steve Callender, G. E. Co.

System Requirements:

Mod 4, 8KB, TTY, BOSS

Language: INTERDATA OS Assembler

Available Material:

Source Paper Tape

This OS Mod 1 Assembler is the standard INTERDATA Model 1 Assembler for Model 3, 4 or 5, 03-034, as modified to operate under BOSS for I/O purposes. Written to run on a Model 4 or 5, it assembles source statements of the Mod 1 Assembler Language as described in the Mod 1 Users Manual, 29-215. Operating procedures are those defined in 03-034A16 Operating Instructions. It uses three logical units which must be assigned as follows prior to execution:

LU1 Source Input Device

LU2 Binary Object Device which must be X'FF' to obtain M08/M09 tape format.

and LU3 List Output Device.

IUG-143: MEDIA TO MEDIA PROGRAM

Author: Steve Callender, G. E. Co.

System Requirements:

Mod 5, 8KB, TTY, BOSS

Occupies:

X'013E' Bytes

Language: INTERDATA OS ASSEMBLER

Material Available:

Source Paper Tape

Description

This Media to Media Program provides the capability, under BOSS, to read ASCII records from one peripheral device to another. Optional features provided are moving the sequence number from columns 72 through 80 to columns 62 through 70, and conversion of Hollerith code produced on either the -029 or -026 keypunch. Extraneous trailing spaces are removed from records. Program options are selected via the settings of the Display Panel's Data Entry Switches' least significant digit.

IUG-144: OS TEXT EDITOR W/IMPROVED OUTPUT

Author: Steve Callender, G. E. Co.

System Requirements:

Mod 4, 8KB, TTY, BOSS

Language: INTERDATA OS ASSEMBLER

Material Available:

Source Paper Tape

This text editor is the standard INTERDATA Text Editor (TIDE), 03-027R01, as documented in the Tide Manual, 29-229 and as improved by the following features:

- 1) Modified to function beyond 32KB;
- 2) O,L,P output of multiple lines made faster;
- 3) High text buffer limit fetched from BOSS as TOP-OF-CORE;
- 4) Reinitialization on restarts of the General Register containing the constant 1;

and 5) Form feed character sent to LU3 for L and P commands.

IUG-145: SOURCE DECK SYSTEM

Author: Steve Callender, G.E. Co.

System Requirements:

Mod 5, 24KB, TTY, BOSS/4B

Mag Tapes

Occupies:

X'F00' to X'43FE'

Language: INTERDATA OS Assembler

Material Available:

Absolute Object Tape

Operating Instructions

This system allows for the creation and maintenance of magnetic tape source libraries where each file, separated by file marks, consists of program Source statements preceded by a TITLE statement and terminating with an END statement. It provides the capability to edit and copy these magnetic tapes. Source files may be listed, extracted, or assembled. In addition, several operator convenience functions are provided.

IUG-146: MODEL 1 MULTIPLY/DIVIDE SUBROUTINES

System Requirements:

Mod 1, 2KB

Occupies: MULTIPLY 59 Bytes

DIVIDE 85 Bytes

Language: INTERDATA MOD 1

Assembler

Material Available:

Source Listing

These two subroutines, Multiply and Divide, provide the Model 1 user with a software unsigned 16 x 16 bit multiplication and a 32 x 16 bit division capability. Execution times are:

MULTIPLY:  $603 + 19N$  usec, N=No. of 1's in multiplier

DIVIDE:  $822 + 986 + 10N$  usec, N=No. of 1's in quotient.

IUG-147: PLOT4 - POOR MAN'S PLOTTER

Author: Richard Sewell, NRC of Canada

System Requirements:

Mod 4, TTY (or line printer)

Occupies:

X'27C' Bytes

Language: INTERDATA OS Assembler

Material Available:

Flow Chart

Assembly Listing

Description

Relocatable Object Tape

This subroutine plots up to four sets of data stored in memory onto printing devices, such as the teletype or high speed line printer instead of a special plotter device.

IUG-148: MAG TAPE UPDATE/MAINTENANCE PROGRAM

System Requirements:

Mod 4, HS Opt., TTY, BOSS  
Card Reader/Mag Tape

Occupies:

X'2A8' Bytes

Language: INTERDATA Basic Assem.

Available Material:

Relocatable Object Paper Tape  
OS Assembly Listing  
Description

This program, under BOSS, allows the user to create, copy and update multi-file magnetic tapes by means of a series of simple commands. Files contain 80 byte records addressable by their rightmost five digits of the sequence field in columns 73-80. Records are re-sequenced on output to LU3 from 00000001 up to 00032767.

Other functions performed by the program include:

TTY Keyboard to Magnetic Tape  
Card Reader to Magnetic Tape  
Card Reader to Printer  
Magnetic Tape to Printer

IUG-149: RANDM - RANDOM NUMBER FUNCTION

Author: Fred V. Brock, Univ. of Okla.

System Requirements:

Mod 4, HS Opt., Flt. Pt.,  
16KB BOSS, Fortran IV, 04-014

Occupies:

X'A2' Bytes

Language: INTERDATA OS Assembler

Available Material:

Relocatable Object Tape  
OS Assembly Listing  
Test Program Listing

This program is an assembly language function subroutine which allows the user to use RANDM as a function in FORTRAN IV programming; such as

```
RX = RANDM(IY).
```

RANDM generates a sequence of 536, 870, 912 random real numbers in the range  $0.4656613E-9 \leq RX \leq 1.0$ ; with population mean 0.5 and variance 0.083333. It is initialized by setting IY to any odd, positive integer  $\leq 32767$  and on subsequent calls IY=0. This function is adopted from the IBM 360/SSP subroutine RANDU and should have the same characteristics.

EXAMPLE: generate 1000 random numbers

```
DIMENSION RX(1000)  
RX(1) = RANDM (7305)  
DO 1 J=2, 1000  
RX (J) = RANDM  
CONTINUE
```

At load time this subroutine object tape should be LINKed to the user's compiled program prior to the EDIT of the Run-Time Library.

IUG-150: RANDU - RANDOM NUMBER GENERATOR

Author: Fred V. Brock, Univ. of Okla.

System Requirements:

Mod 4, HS Opt., Flt. Pt., 16KB

BOSS, FORTRAN IV, 04-014

Occupies: X'64' Bytes

Language: INTERDATA OS ASSEMBLER

Available Material:

Relocatable Object Tape

OS Assembly Listing

Test Program Listing

This program is an assembly language subroutine which allows the user to call RANDU in FORTRAN IV programming, such as:

```
CALL RANDU(IY,IX,RX)
```

RANDU generates a sequence of 16383 random integers, IX, and real numbers, RX. The range of these numbers is  $1 \leq IX \leq 32767$ ,  $0.305185E-4 \leq RX \leq 1.0$ . The distribution is uniform over this range with population mean of 0.5 and variance 0.083333. On the initial call to RANDU, IY must be set to any odd, positive integer  $\leq 32767$ , and on subsequent calls, IY must be set equal to the previous value of IX. This subroutine is taken from the IBM 1130/SSP.

EXAMPLE: generate 1000 random real numbers.

```
DIMENSION RX (1000)
CALL RANDU (7305,IX,RX(1))
DO 1 J = 2,1000
CALL RANDU (IX,IX,RX(J))
1 CONTINUE
```

At load time, this subroutine's object tape should be LINKed to the user's compiled object tape prior to the EDIT of the Run-Time Library.

IUG-151: VAPOR PRESSURE - TEMPERATURE TABLE

Author: Wm. Shulman,  
Edgewood Arsenal, Md.

System Requirements:

Mod 3, TTY, 16KB

Interactive FORTRAN w/Trap  
or Mod 4 with either Int.

FORTRAN SW/Flt.Pt. or  
Int. FORTRAN HW/Flt. Pt.

Language: Interactive FORTRAN

Available Material:

Source Listing (as created  
by 03-006R02)

Source Tape (as created by  
03-006R02)

Test Program Listing (using  
Methanol at low temp)

Description

The purpose of this program is to give a tabulated list of temperature vs. vapor pressure based on either 2 or 3 laboratory generated points. The program will take either Centigrade or Fahrenheit temperature and either millimeters, mercury or atmospheres pressures (not inches). The program solves the equations:

$\ln(VP) = B + A/T$  (ABSOLUTE) for the two point case  
and  $\ln(VP) = C + BT$  (ABSOLUTE) +  $A/T$  (ABSOLUTE) for the three point case.

When given a lower and upper value and increment, the program types a table of values in millimeters, mercury and atmospheres.

IUG-152: SIMULTANEOUS EQUATIONS SOLVER

System Requirements:  
(Developed on Model 14,  
32KB using a Time-shared  
version of Interactive  
Fortran)  
Any interactive fortran  
with associate configuration

Language: Interactive Fortran  
Available Material:  
Source Paper Tape  
Source Listing

The purpose of this program is to solve a set of up to 14 simultaneous equations. The technique used is the Gauss Elimination as presented in varied mathematical texts and/or matrix methods. The source paper tape contains both the main subroutine Gauss Elimination (GE) and Absolute Value (AV) which is called by AV. To solve more than 14 simultaneous equations the arrays X and A would have to be re-dimensioned.

IUG-153: MAG TAPE GENERAL LOADER

System Requirements:  
Model 3/4 HS OPT (RB)  
REL/GEN Loader (R01)

Language: Interdata OS Assembler  
Available Material:  
Relocatable Object Tape  
(M16) Assembly Listing

This program is similar to the Interdata General Loader, 06-025R01, as described in Loader Descriptions, 29-231. It is, itself, loaded by the REL or GEN Loader (R01 versions).

IUG-153 loads 108-byte fixed-length object data records from 9-track magnetic tape. The records must be in standard Interdata non-zoned loader format without the leading X'F0' character found on M16/17 paper tape formats. For example, object records output under BOSS 4A/4B with a binary write SVC are loadable by this loader. It does not load object paper tapes. No tape positioning is available. The Device Definition Table BINDV at location X'78' must be set to X'YY00' where YY=Magnetic Tape Device Address, e.g. X'8500' or X'9500'.

On parity error, eight re-reads are attempted resulting in a "READ ERROR" message if receipt of the ERR status bit persists. The message "EOF READ" results from an EOM status returned, and the "DEV END" message results from an EOT status returned.

IUG-154: BOSS/4A INTERTAPE DRIVER

System Requirements:  
Model 4, 16KB, BOSS/4A,  
Intertape  
Occupies: X'23' Bytes

Language: Interdata OS Assembler  
Available Material:  
Relocatable Object Tape  
Source Listing  
Linking instructions

The Driver provides all of the functions available in the BOSS/4B Intertape Driver, Interdata Part Number 07-042F02. These functions incl

Read, Write, Backspace, Skip Record, Write an EOF, Skip Forward to EOF, Skip Reverse to EOF, and Rewind. The driver is subject to the same restrictions and considerations as the BOSS/4B Driver mentioned above. In developing this Driver, the logic of the BOSS/4B Driver was modified and augmented to take full advantage of the I/O Proceed and Task Queuing facilities of BOSS/4A. Although the linking instructions overlay the HSPRT/P, LP, CR, and MT Drivers with this Intertape Driver; similar system commands may be used to locate it elsewhere in memory with the user establishing his particular driver entry points in AETAB and adjustments to UBOT of BOSS/4A.

IUG-155: INTERDATA MODEL 70 CROSS-ASSEMBLER FOR USE ON IBM 360/67  
CP/CMS

System requirements:  
IBM 360/67 CP/CMS, 65KB

Language: 360 BAL  
Available material:  
6 Assembly listings  
6 Object decks  
(on special request)

This program is a cross-assembler for the Interdata OS Assembler (Model 70) on the IBM 360/67 running under Model 5 Assembler, IUG-126; with reference to Interdata OS Assembler, 03-025. The cross-assembler consists of five programs and allows a flexible choice of output methods. Also, any embedded "AUTO PLOT" statements (looking like comment cards to the assembler) can then be used to create flow charts on the IBM 360.

IUG-156: TELETYPE TEST PROGRAM (TELTYP)

System requirements:  
Model 4, 8KB, TTY  
Occupies: X'2E2' Bytes  
Using 72 byte Buffer at  
X'760' absolute

Language: Interdata Basic Assembler  
Available material:  
Relocatable object tape  
Basic assembly listing  
(no comments)  
Operating instructions

This is a teletype test program similar to the Interdata Teletype Test Program 06-004R03. It differs in that the program is relocatable and prior to character input it types the message "ENTER CHARACTERS".

IUG-157: STORAGE OSCILLOSCOPE TEST (STOSCO).

System requirements:  
Model 4, 8KB,  
Interdata Scope Interface

Available material:  
Relocatable Object Tape (M08)  
Basic Assembly Listing  
Operating instructions

This program produces a pattern on the oscilloscope face that is output just once in order to test storage scopes. The pattern is a grid of evenly spaced parallel orthogonal lines. It can be used to test the linearity and uniformity of the scope. Setting Display Panel Switch 15 causes the program to repeat the scope display continuously.

IUG-158: CORE I/O SIMULATOR DRIVER (BOSS/4A)

System requirements:	Language: Interdata OS Assembler
Model 5/70, 24KB-40KB,	Available material:
BOSS/4A, REL/GEN Loader R01,	Source Paper Tape
OS Tide, OS Assembler or	Assembly listing
FORTTRAN IV	Relocatable Object Tape

Occupies: X'4A' Bytes

This program is designed for use on machines with large memory storage. This driver allows source input for the OS Assembler or FORTRAN IV to be read directly from a pre-loaded "Tide" Buffer. This allows alterations and recompilations to be repeated with no need for source I/O operations. The driver is loaded according to standards, adjusted UBOT and "AETAB" in Boss/4A patched to reflect a Dummy Physical Unit, and Driver Address. Halfword X'48'R in the driver must be patched to point to the start of the OS Tide buffer.

The desired Logical Unit for source input is assigned to the Dummy Physical Unit. A rewind on this unit must be used to re-initialize the Tide buffer for each compilation or assembler pass.

IUG-159: FAST FORMAT PUNCHER FOR INTERTAPE

System requirements:	Language: Interdata Assembler
Model 3, 8KB, TTY,	Available material:
Intertape, IUG-160	Relocatable Object Tape

Occupies: X'01BE' Bytes  
Source Listing

This program is an adaptation of the Fast Format Puncher 06-031. The output routine has been rewritten to accommodate Intertape. The copy option of 06-031 has been deleted. The program dumps a block of memory bounded by user-entered addresses at X'0000'R (Low Address) and X'0002'R (High Address). The data is dumped on BOUTDV. The user may specify a transfer address in X'0004'R. The program must be loaded with the General Loader, 06-025R01. It is started at X'0010'R. The program has been used successfully in conjunction with the Fast Format Loader for Intertape, IUG-160 to create Cassette Tape Bootstrapped modules of BOSS/4A and BOSS/4B.

IUG-160: FAST FORMAT LOADER FOR INTERTAPE

System requirements:	Language: Interdata Assembler
Model 3, 8KE, TTY	Available material:
Intertape	Relocatable Object Tape

Occupies: X'80'-X'C2'  
X'D0'-X'136' and  
X'1FA' Bytes Below  
Load Bias  
Source Listing

The program is an adaptation of the Fast Format Loader, 06-030R01. The loader input routine was modified to accommodate the Intertape.



Mainly, it implements back-spacing and re-reading in the event that ERR (Intertape Status Bit 0) = 1. The program must be loaded with the General Loader, 06-025R01, with the Bias Definition Value (X'NAOA') set to the address which the last byte of the Fast Format Loader is to occupy. Thus, if [X'NAOA']=X'3000', the Fast Format Loader will occupy X'2E06'-X'3000' upon loading.

The General Loader terminates with the PSW set to X'80000 00D0'. Depressing EXEC (starting at X'D0') causes a file gap to be written on BOUTDV followed by an 8-bit image of a boot loader which can be loaded by the 50-Sequence, followed by a checksummed 8-bit image of the Fast Format Loader.

Used in conjunction with the Fast Format Puncher for Intertape, IUG-159, Bootstrapped load modules may thus be created. The Model 4 50-Sequence and Device Definition Table must be modified as follows (to load this Boot and its subsequent program/data dumped in Fast Format):

	50	D500	AL X'C2'
	52	00C2	
	54	4300	B X'80'
	56	0080	
BINDV	78	55C1 or 45C1	
BOUTDV	7A	55C2 or 45C2	

Note that in order to use this 50-Sequence, the tape must be manually positioned off the BOT clear leader.

Note: If the Fast Format Puncher is to be used to dump a block of memory following the header output by this program, DO NOT REWIND THE TAPE.

#### IUG-161: FILER INDEX LISTING PROGRAM

<u>System requirements:</u>	Language: OS Assembler
Model 74 Instruction Set,	Available material:
RTOS or BOSS, TTY	Relocatable Object Tape
List Device, Mass Storage	OS Assembly Listing
Device	Description

This program produces a listing, on the Device Assigned to Logical Unit 3, of the named files on a mass-storage Logical Unit 4 which has been maintained by the Filer, Interdata Program 07-047. The listing shows the name of the file, the address of the first record of the file, and the number of records to add to the file when it is expanded. The program also prints the number of unused sectors left on the Logical Unit containing files.

IUG-162: BOSS CORE DRIVER

System requirements:

BOSS, OS Tide, OS Assembler  
or FORTRAN IV Compiler  
Model 5/70, 24KB

Language: OS Assembler

Available material:

Source Paper Tape  
Relocatable Object Tape  
Source Listing  
Assembly Listing

This program allows users with large memories to have either the OS Assembler or FORTRAN IV Compiler resident at the same time with OS Tide under an operating system, BOSS. This provides users with a combination assembly-editor or compiler-editor. Assigning a dummy physical address to the OS Tide text buffer in core, and also assigning Logical Unit 1 (LUL) to this dummy physical address causes the OS Assembler or FORTRAN IV Compiler to read source directly from OS Tide. This includes multiple passes when required. To link to BOSS, UTOP may be adjusted to include the driver and AETAB must be adjusted with the address of A(CORED V) of the driver for physical device "FE". The driver must be given the selected limits of the text buffer in OS Tide and the addresses of ILFUNC and ZZZZ of BOSS/4B.

IUG-163: INTERTAPE TEST PROGRAM FOR MODEL 4

System requirements:

Model 4, 8KB  
Intertape M46-400

Language: OS Assembler

Available material:

Relocatable Object Tape  
Assembly Listing  
Description

Occupies: X'0D76'R Bytes

This program is an updated version of the Interdata Cassette Tape Test Program 06-110, which has been rewritten to be used with Interdata's Intertape Product M46-400. Its intent is to provide the user with a test program which occupies less than 8KB memory. It contains facilities to check all features of the Intertape system by switch options.

IUG-164: INTER-SIG 707

System requirements:

Model 70, 8KB, TTY  
10-023F Data Set Adapter  
103 Data Set, Full duplex  
time share service capable  
of suppresssing the echo  
from the keyboard.

Language: Interdata Basic Assembler

Available material:

Relocatable Object Tape  
Assembly Listing  
Operating instructions  
Description

This program allows one to converse with the Xerox UTS or any full duplex time sharing service using a TTY console device, Model 70 processor, full duplex data set adapter and a 103 Data Set. The program occupies X'52C' Bytes.

IUG-165: 2-BYTE INTEGER FORTRAN IV COMPILER AND RUN-TIME LIBRARY

System requirements:

Model 4, 16KB, TTY  
Floating Point

Language: OS Assembler

Available material:  
Relocatable Object Tapes  
Description

Source listings or tapes are not available for the 2-Byte Compiler. Users must have the FORTRAN IV software package, 04-014 to obtain additional documentation and the remaining run-time library tapes. The modification allows the "two-byte" integer to be used in any FORTRAN IV statement in which the "standard" integer may be used. The only difference is that two bytes of core storage are allocated for the standard integer. The two types of integers will be functionally equivalent, except in those areas affected by the difference in core allocation.

IUG-166: BULLSEYE - THE ARTILLERY GAME

System Requirements:

Model 70, 16KB, TTY

Available material:

Source Paper Tape created  
by Interactive Fortran

Bullseye is a game demonstrating the interactive nature of some programs. It asks for a target distance in miles, elevation angle in degrees, and velocity of the projectile in feet/sec (i.e. 1.5, 45,600 (CR) ). Remarks are made to goad the player, and the distance undershot or overshoot is printed in feet. The user then updates the angle and/or velocity. There is a replay option.

IUG-167: YR - 20TH CENTURY DAY-OF-THE-WEEK GENERATOR

System requirements:

Model 70, 8KB, TTY

Available material:

Source Paper Tape created  
by Interactive Fortran, Flow chart

This program requests input of date, month, year. Input is from TTY 6-digit numerical representation of any date in the 20th Century. Output is "check n" where n is number of days into 20th Century of input, then the day of the week on which that date fell.

IUG-168: OSOBAN - OS OBJECT TAPE ANALYZER

Submitted by: William Vaughan  
Interdata, Inc.

System requirements:  
Model 4, 8KBm BOSS/4A or BOSS/4B,  
DOS or BOSS

Language: OS Assembler

Available material: Object tape, listing, description.

This program analyzes a standard Interdata object tape in M08, M09 or M17 format. The analysis is printed in a descriptive, easy-to-read format. Invalid loader items are detected. No error checking is performed on sequence number and checksums.

IUG-169: DISK FILE SYSTEM

Submitted by: J.T. Beckett  
Bell Telephone Labs.  
Naperville, Illinois

Language: OS Assembler

System requirements:  
Model 4, SELCH, TTY, Disk

Available material: Source or object tapes, description, listings

This program was designed for an Interdata Model 4 with Diablo Disk. It permits the allocation of named files and the transfer of data in absolute form between the disk and specified core limits. The program resides at the top of memory to permit loading programs such as the BOSS operating system and other Interdata test programs which must reside in low core. Accompanying programs provide a bootstrap facility and the means for dumping the disk onto another device for backup protections.

IUG-170: ORI EDITOR

Submitted by: T.J. Monks, NASA, Greenbelt, Maryland.

System requirements: Model 5

Language: OS Assembler

Available material: Object tape, description, listing

This Editor program has been implemented for the Interdata Model 5 to run under the Interdata Basic Operating System. In order to create an editing tool closely resembling the existing Interdata Text Editor (TIDE), some changes were made in the nature of names and commands and order of arguments. In this way it would be possible for a user to use the ORI Editor without realizing he is not using TIDE, since EDITOR functions are a superset of TIDE functions.

IUG-171: INTEGER SQUARE ROOT SUBROUTINE FOR MODEL 5 & NEW SERIES

Submitted by: J. Spooner, General Electric Company  
Waynesboro, Va.

System requirements:  
Model 4, 8KB, TTY

Language: OS Assembler

Available material: Object tape and listing

This subroutine extracts halfword root from fullword argument with optional rounding of result. Execution time is considerably less than for method of successive approximations. Model 70 average time is 400 microseconds.

IUG-172: M10 TAPE GENERATOR (PAPER TAPE DUPLICATOR IN M10 FORMAT)

Submitted by: J. Spooner, General Electric Company,  
Waynesboro, Va.

System requirements:  
Model 4, TTY, HSPTR

Language: OS Assembler

Available material: Object tape, operating instructions

Simple to use program creates M10 bootstrappable tapes which can be loaded by Model 5 bootstrap. Program requires X'3AC' bytes in addition to that occupied by program for which tape is to be generated.

IUG-173: CHARACTER TIME

Submitted by: Scott Hughes, Atmospheric Environment Service,  
Toronto, Canada

System requirements:  
Model 4, 8KB, TTY, Loader

Language: OS Assembler

Available material: Source tape, program description, operating instructions

The purpose of this program is to allow ease in adjusting the timing of devices on the multiplexor bus. Device address and desired transmission rate are entered on the console switches and an indication of how close the desired and actual transmission rates are, is displayed on the console lights.

IUG-174: BOSS 4A BOOTTAPE CREATOR

Submitted by: Scott Hughes, Atmospheric Environment Service,  
Toronto, Canada

System requirements:    Language: OS Assembler  
  Model 4, 8KB, HSPTP, BOSS/4A  
  Rel/Gen/BOSS or OS Loader

Available material: Source tape, assembly listing, program description,  
operating instructions

This program punches a bootstrap tape of the current version of BOSS/4A in core. The bootstrapping process used on the tape does not modify core outside of BOSS by placing the M10 format loader in the Interrupt Service Table. Using this program, drivers or patches may be added to BOSS, and a bootstrapping tape can be made of the version.

IUG-175: OS INT 4TRAN RW EXPANSION

Submitted by: R.J. Paoli, Space Sciences Laboratory  
University of California at Berkeley

System requirements:    Language: OS Assembler  
  Model 4, 16KB, HS Opt.,  
  Floating Point, TTY.

Available material: Source/object tapes, listing, description, operating instructions.

This program allows the use of all peripheral devices that are available to the Operating System to be used as logic units with the READ-WRITE statements in Interdata's OS Interactive Fortran, 03-033. The program is loaded with the OS Loader. It includes an initialization procedure which provides the Fortran linkages, adjustment of the user buffer and program transfer to the Interactive Fortran.

IUG-176: OS BLACK JACK

Submitted by: Wayne Lowzik, Interdata, Inc.

System requirements: Model 5 or New Series,                            Language: FORTRAN IV  
  16KB, Floating Point, TTY, BOSS, DOS

Available material: Source/object tape, listing, description, operating instructions

This program runs standard Black Jack (21) under an operating system. The program uses logical unit 5 and all operatin input comes in from the TTY. The deck of cards are on paper tape supplied with the program. Deck is shuffled in core. The program asks that the deck be assigned LU4.

IUG-177: YAHTZEE - A GAME PROGRAM IN BASIC

Submitted by: Philip Stein, National Bureau of Standards, Washington, D.C.

Available material: Source paper tape and listing

Yahtzee is a game for one to four players. After introducing the players to the computer, five dice are rolled and their face values printed. As in draw poker, the player has two changes to improve his hand by ROLLING any or all dice. The object of the game is to score each improved hand on one of thirteen "lines" on the score sheet. Each line may be used only once, and when all lines are used, the game is over.

IUG-178: GOLF IN BASIC

Submitted by Philip Stein, National Bureau of Standards, Washington, D.C.

Available material: Source paper tape and listing

This program plays golf with the user. The user chooses clubs and how hard to swing. The source is adjustable in data statements.

IUG-179: ONE-ARM BANDIT GAME IN BASIC

Submitted by Philip Stein, National Bureau of Standards, Washington, D.C.

This program plays the slot machine game One-arm Bandit with the user.

IUG-180: DATAPLOT 70

Submitted by: Carol Young and Philip Stein, NBS, Washington, D.C.

System requirements:

Model 5 or New Series, 16KB,  
Electrostatic plotter.

Language: OS Assembler, FORTRAN IV

Available material: Object tape, compiler listing, example printout, test listing.

Dataplot 70 is a Fortran Sine-wave test program. It is a complete package for converting point-by-point data to raster form for an electrostatic plotter. Program includes FORTRAN subroutines for scaling data, labelling and drawing axes, and drawing titles.

IUG-181: OS DELOADER

Submitted by: Richard Freemire, National Bureau of Standards, Washington, D.C.

System requirements: Language: OS Assembler  
Model 5/New Series, 8KB, and BOSS  
or other OS source and listing device.

Available material: Object tape, source listing and example printout

OS Deloader reads M08 or M16 object programs (like OSOBAN, IUG-168) and produces: (a) a listing with loader control items and assembly op codes and formats. An attempt to fit ASCII printing characters if also made for each line; (b) a disassembled, relocatable output which can be re-assembled.

IUG-182: MAGTAPE DUMP

Submitted by: R. Crabtree, Atmospheric Environment Service,  
Toronto, Canada

System requirements: Language: OS Assembler  
Model 70, 8KB, TTY, HSPTR/P, and a  
binary input device.

Available material, Source/object paper tape, operating instructions

This program allows the user to investigate mag tape records of varying lengths in a hexadecimal format. The program reads in logical records from a binary device, unpacks the data and outputs the hex character string to a character (ASCII) device.

IUG-183: MAGLDR

Submitted by: D.R. Hicks, RCA, Camden, New Jersey

System requirements: Language: OS Assembler  
Model 70, 8KB, TTY, HSPTR,  
Mag Tape Cassette

This is a patched version of the General Loader (06-025R01) to permit finding labeled programs, and to permit mag tape operations.



IUG-184: BOSS/4D

Submitted by: D.R. Hicks, RCA, Camden, New Jersey

System requirements:

Model 70, SELCH, TTY, Mag Tape

Language: OS Assembler

Available material: Object tape, assembly listing, description, operating instructions.

This program is a patched version of BOSS/4A which permits commands of the form "LOAD LU Name" so that programs on a mag tape library can be found and loaded. The program also permits commands to change PSW and reset default system parameters. HSPTR/P, LP, and CR drivers have been deleted.

IUG-185: EBCDAS (Subroutine to convert an EBCDIC character into its ASCII equivalent)

Submitted by: Roy Stehle, Stanford Research Institute, California

Language: Interdata 70 Fortran

FORTRAN Call Statement: CALL EBCDAS (KHAR)

This program converts the 8-level EBCDIC character KHAR into a 7-level ASCII character which is returned in KHAR. The program first tests the value of KHAR. If KHAR is less than zero or greater than 255, KHAR is returned unchanged. The program uses table look-up to convert the 64 character FORTRAN set as well as 25 control or symbol characters. In some cases, an illegal EBCDIC character will be converted into an ASCII "@".

IUG-186: GPM - GENERAL PURPOSE MACRO ASSEMBLER

Submitted by: E.P. Estes, Interdata Inc.

The General Purpose Macro Assembler, GPM, is a symbol stream processor designed to run on an Interdata model 4, 5 or New Series minicomputer. The program must be run under one of the Interdata operating systems. The program processes an input symbol stream which is read from logical unit 7. The input symbol stream may consist of: 1. Macro definitions; 2. Macro calls; 3. Transparent input symbols.

IUG-187: PLOT 10 - ACCESS LEVEL SOFTWARE & GRAPHICS DEMO

System requirements: Interdata Models 3, 4, 5 or New Series, 8KB of core, paper tape reader, 4010 Computer Display Terminal with a TEKTRONIX Teletype Port Interface.

Access Level Software (ALS) has been written to facilitate the use of the TEKTRONIX 4010 Computer Display Terminal with the Interdata computers. The cursor is supported in addition to the terminal. The software includes basic routines for the following:

- Plotting
- Graphic Input
- Input of any ASCII character
- Output of any ASCII character

All routines are written in assembly language.

IUG-188: FAST FOURIER TRANSFORMS ON INTERDATA PROCESSORS

System requirements: A processor capable of running on Interdata general purpose operating system (BOSS, DOS or RTOS). The routines are written in Kearney & Trecker FORTRAN and assembler. (Note: the compiler may not execute on a Model 74, this will require another processor for system maintenance.)

Operation in 8KB of core may be possible with a subset of facilities, but in general a 16KB processor will be required.

All operations are performed as 16-bit two's complement integers. However, the input data should be confined to the range 16383 to -16384. A block exponent technique is used to relieve the user of the problem of scaling data. The standard version supports a maximum of 1024 points, but this may be extended by changing 'STABLE'.

IUG-189: SELECTOR CHANNEL READ/WRITE PACKAGE

This is an assembly language program with a series of Fortran-callable entries for the Interdata New Series Model 70. This package, which interfaces with the magnetic tape unit, is written in relocatable loader format and stored on the library unit (AO) with the label WRDATA. It includes four Fortran-callable entry points:

- WRDATA
- RDDATA
- IWRSTA
- LENGTH

IUG-190: SYMBOL - Text-printing Using Interdata 70 and TEKTRONIX 4010

SYMBOL is a demonstration of a text-printing program using the Interdata Model 70 and the TEKTRONIX 4010 display terminal. The distinctive feature of this program is that it does not use the hardware character generator included in the 4010. Rather, it draws the characters using a series of vector line segments pre-stored in the computer memory. One advantage of this technique is that the character fonts can be modified to fit the user's requirements solely by software changes.

This particular character set has been designed to match the standard Leroy drawing templates widely used for report illustrations. At present, the set includes a full upper and lower case plus a complete Greek letter set. It also contains many of the commonly used punctuation marks and special characters. The program is FORTRAN callable.

IUG-191: SREAD - FORTRAN-callable Subroutine

This routine is used to permit free-form entry of data to a program. The number of data inputs is specified by NUMBER. The logical unit which will be interrogated is LU. When the specified number of data inputs has been read from the logical unit named, the data values will be returned in a real array named ARRAY. This array should be at least as large as the largest value of NUMBER used in the main program.

Data input may be in integer, fixed-point, or scientific notation. The data returned to the main program will always be real, and will be an approximation of the input value.

IUG-192: TIMR86

TIMR86 computes the execution time of a program on the Model 80 or Model 60. It runs under any operating system and uses LU1 for source input, LU3 for listing output.

Every time it sees a label it begins accumulating a new counter of execution time. Every time it sees a conditional branch it prints out all the accumulated times. When it sees an unconditional branch, it starts with a single counter. It will automatically compensate four times when branch is taken or not taken. For load and store, multiply, and shifts, it assumes an average execution time.

IUG-193: TIMR75

This program is exactly like the TIMR86 except that it computes the execution time of a program on the Model 70 and Model 50.

IUG-194: ANALAD

This program, written in FORTRAN IV, will analyze circuits with a "ladder" topology, i.e. alternating series and shunt elements. The circuit can be made of RLC networks. The size of the circuit is not restricted, only the topology. The program solves the ABCD matrix of the network.

IUG-195: CLEAR

This routine, in Interdata assembly language, sends the Acknowledge Interrupt instruction until a time-out status is returned. The routine returns with the last device address received in the argument NDEV, which should be zero.

IUG-196: COMAND

This program, written in Interdata assembly language sends an output command to device NDEV with bit pattern found in lower 8 bits of NCMD. On return, lower 8 bits of NSTAT contain status read from device. If NCMD equals zero then no command is sent but current status of NDEV is returned in NSTAT. Execution of this routine requires that the Protect Mode is disabled.

IUG-197: DFIX

This re-entrant routine takes the double precision value in X and converts it to an unnormalized binary number and stores the most significant 32 bits in the array Y.

IUG-198: EOFIND

Entry of SKPEOF reads forward on logical unit LU until an EOF mark is found. Entry BKSEOF reads backward on logical unit LU until an EOF mark is found.

IUG-199: EOF85

This program contains a series of entry points that enable the Fortran programmer to manipulate the magnetic tape unit directly without logical unit assignments through BOSS. The 8-bit status returned from the mag tape device is returned in the low-order bits of the argument ISTAT.

#### IUG-200: EXISER

On external interrupts (if external interrupts are enabled) the computer will stop currently-running program and branch to this routine with external interrupts disabled. Routine first sends an interrupt acknowledgement and determines number and status of interrupting device. It then determines a branch entry point address for the device (by table lookup), sets up Fortran linkages, places the address of the memory cell containing the current status of the device in the first argument position (in integer format), and branches to the appropriate entry point.

On return from the device service program, the EXISER routine again sends an interrupt acknowledgement. If no interrupts are waiting, the routine returns to the interrupted program.

#### IUG-201: KHAR FUNCTION ROUTINE

This function routine reads one character from the teletype keyboard (device 02) and returns with the character read in the upper 8 bits of KC. The routine does not use the BOSS driver. ID is a dummy argument. Routine requires the Protect Mode bit of the PSW status word to be reset.

#### IUG-202: MOVEB ROUTINE

This assembly language routine extracts N bits from the full 32 bit word location A, starting at bit position L1, and inserts these bits into word B, starting at location L2. Bit positions are designated from right to left starting with bit position 0.

#### IUG-203: MOVEIO

This program permits the Fortran programmer to convert integer and real numbers to ASCII characters internally via calls to the formatted I/O routines, @I, using a standard Fortran format statement.

#### IUG-204: MASKA ROUTINE

This routine masks off the Nth bit position of the contents of the full 32-bit word A. If the bit designated by N in A is "1" the value of the function is returned as (-1). If the bit is zero, the value of the function is returned as false (0). Bit positions are designated from right to left with the least significant bit position designated bit 0

IUG-205: OVLAY

This program contains a series of Fortran-callable entries that enable the Fortran programmer to overlay programs in memory from an external storage device and start execution. The program utilizes a modified OS Library Loader program (label LOADR1) which must be resident in memory with this program.

IUG-206: PROC IN INTERDATA FORTRAN

Program converts Univac 1108 system 6-level BCD code found in argument IB to equivalent 7-level ASCII code and stores in internal 68 character buffer.

IUG-207: PSWMOD IN ASSEMBLY LANGUAGE

This program contains a series of Fortran-callable entry points that enable the Fortran programmer to change the current PSW status in real time.

IUG-208: SHIFT

This program shifts the contents of location IA by N bit positions and returns the value in I. The contents of IA are not changed. If N is positive, the shift is left circular. If N is negative, the shift is right end-off with sign extension.

IUG-209: SFTENT

This routine takes the contents of the full word A, and left shifts (end-off) 8 bits. It then adds the lower eight bits of the half-word IBYTE to the lower 8 bits of A and places the result in B. The contents of A and IBYTE are not changed.

IUG-210: SIGATN

Program takes the integer value in argument NDB and converts to three BCD digits. It then transmits these digits in ascending sequence to the temporary storage register via the Universal Interface Module (device 9B). Finally, a command to transfer the contents of the temporary register to the ATTN register is sent. Routine returns after the interface returns a "not busy" status. Program requires the Protect Mode bit of the PSW status word be reset.

IUG-211: SIGFRE

Entry SIGFRE takes the first ten integer decimal digits found in the double precision argument FREQ and converts them to BCD format. It then transmits these digits in ascending sequence to the temporary storage register via the Universal Interface module. Finally, a command to transfer the contents of the temporary register to the CF register is sent. Routine returns after the interface returns a "not busy" status.

Entry SIGBCD assumes the argument (NCD) has previously been un-normalized by subroutine DFIX (IUG-197).

IUG-212: SOURCE PROGRAM LIBRARY MAINTENANCE PROGRAM

The Mag Tape Source Library Maintenance program allows users to create, copy, and update multifile ASCII magnetic tapes by means of simple commands. The object program is in relocatable form and is located on the program library with label SOURLI.

LU0 = Command device  
LU1 = Input tape  
LU2 = Output tape  
LU3 = Insert device

The program starts by rewinding the input tape and output tape. It advances past the first EOF on the input tape and writes a double EOF on the output tape and backspaces over the last EOF. It then returns to the command mode and indicates this by printing the character '#' on the command device. Starting the program at ORG + IC will not reposition the input or output tapes but will write a double EOF on the output and backspace over the last.

IUG-213: GAMES1

This program is actually three separate demonstrations using the console display and switches:

1. Start at ORG to display switch settings and their complement.
2. Start at ORG+X'20' to have binary counter with speed set by console switches.
3. Start at ORG+X'40' to display  $(2^{**}16)-1$  pseudo-random sequence.

#### IUG-214: MAG TAPE DEADSTART LOADER

Loads an absolute block of program code in 8-bit format into core memory from Mag Tape device 85. (code loaded is assumed to be a relocatable mag tape general loader program). This bootstrap program is designed to be loaded with the 50 sequence from the paper tape reader starting at location X'80'. Program sends X-OFF to tape reader, rewinds mag tape device at location X'7000'. On parity error, program rewinds and rereads. After error-free load, program branches to location X'7000'.

#### IUG-215: CROSS REFERENCE ASSEMBLER

This Cross Reference Assembler operates basically the same as the standard Interdata OS Assembler. During the printing pass the statement numbers are printed between the object data and source statement on the listing. At the end of the pass the standard symbol table is printed along with the statement numbers of instructions which references the symbol.

This assembler requires one additional logical unit assignment for the Cross Reference file. The file must be on a disc and should be approximately 144 sectors (3 cylinders) for a source deck of approximately 2000 cards. No change has been made to the OPTION card. The list device must have 132 print positions. If the teletype is used for the listing some of the references may be lost.

#### IUG-216: ARTILLERY GAME IN BASIC

This program is essentially the IUG-166 rewritten in BASIC

#### IUG-217: GOOK1 & GOOK 2 IN BASIC

These two programs will assist in the generation of "gobbledegook" required in the compilation of reports.

#### IUG-218: GUESS WHAT? IN BASIC

The computer will select a number of up to 4 digits. You are asked to guess the number. After each guess you will be told the number of correct digits and the number of digits in the correct position. 14 chances to guess the number are permitted.



IUG-219: INTERDATA-IBM 360/370 CROSS ASSEMBLER (Requires user supplied MT)

This program assembles programs written in standard Interdata source language, producing object code for the Interdata processor. The Cross Assembler requires an IBM 360/370 under either OS MFT or OS/MVT and at least a 64K region size. Object code from the Cross Assembler is provided in two formats written to data sets SYSPUNCH and SYSCARD. The SYSPUNCH data set refers to a partitioned data set in which 108 byte records in standard Interdata relocatable object program format are written. The SYSCARD data set is sequential data set in which the 108 byte relocatable object program records are placed after being reformatted so that 2 object records become 3 80-byte records. This data set is then suitable for punching on cards.

IUG-220: FORTRAN CROSS ASSEMBLER (DOCUMENT ONLY)

This is a FORTRAN computer program to assemble the machine code for the Interdata Model 70 on the Control Data 3800.

For installations that have access to a large centralized computer, it is frequently advantageous to assemble the machine code for a minicomputer on the large computer rather than purchasing additional core for each minicomputer for the assembly routine.

The conversion of the mnemonic code to the machine code is done in three steps. First the operational code and the operational registers are established together with the instruction locations. Parallel to this, the variable and constant lists are filled with temporary locations assigned if no fixed locations were specified.

In the second step, the ambiguity of the variables, constants, and locations is checked, the final locations of the variable and constant list are determined, and the external list is set up.

In the third step, the symbolic names are converted to machine code, arithmetic conversions of locations are performed, and the index registers are inserted. Finally, a summary of all symbolic names and instructions is printed.

A fully documented report of this program may be obtained from the U.S. Government Printing Office, Washington D.C. 20402 and is designated NOAA TR ERL 269-APCL 28.

IUG-221: OS XRFMT

OS Assembler Aid In FORTRAN IV

IUG-222: 23 MATCHES GAME IN BASIC

IUG-223: BLACK JACK GAME IN BASIC

IUG-224: CHARACTER MANIPULATION & COMPARISON ROUTINES

Four serially reuseable and Fortran callable subroutines:

KOMPAR - Compares two full words

STRCMP - Compares two strings of any given length

INSERT - Inserts a character into a string of characters

EXTRAC - Extracts a character from a string of characters

IUG-225: URAND

A serially reuseable, Fortran callable subroutine which generates pseudo random numbers from a uniform distribution.

IUG-226: DOS DISCTAN

A disc utility to run under DOS, which allows the transfer of data between main memory and disc upon operator requests from the keyboard device.

IUG-227: LINE FREQUENCY CLOCK DRIVER

A modified version of the standard RTOS Line Frequency Clock Driver. Modifications provide month and year advancing logic and temporarily disable external interrupts while the time-out count for any device is being manipulated.

IUG-228: DOS BOOTSTRAP LOADER (FOR FIXED-HEAD DISC)

Modified version of Interdata RTOS Bootstrap Loader. It reads a core image from the disc into memory and assumes the starting point to DOSCYL, the cylinder address of DOS.

IUG-229: RTOS DISCTAN

A disc utility to run under RTOS which allows the transfer of data between a one sector sized buffer (256 bytes) and the disc, upon operator requests from the keyboard device.

IUG-230: LUPA/GETPA

Two Run-Time Library subroutines (16-bit machines only)

LUPA - Logical Unit to Physical Address  
GETPA - Obtain Physical Address IPA currently assigned  
to Logical Unit LU

IUG-231: DIABLO DISK DRIVER FOR MODEL 4 AND BOSS/4B

This program is a modification of IUG-123, Interactive Fortran Read-Write Functions.

IUG-232: "UTILIZING A FIXED HEAD DISC UNDER DOS"

A write-up on changes to the Interdata Disc Operating System (DOS) to allow support of a Pacific Micronetics fixed-head disc. The fundamental differences between the Diablo disc driver, the standard DOS, and the PM disc driver are described. These changes may be applicable to the support of other fixed-head discs or drums having the same command and status bytes, sector size (256 bytes) and sector addressing techniques.

IUG-233: BASIK - 132 CHARACTER BASIC (9 ZONES)

A modification of Interdata's BASIC Interpreter to permit printing 132 character records and to tabulate to 9 zones with the ninth zone extending to 132 characters. This program is not for use with Interdata standard MUE. Use with BOSS, DOS, etc.

IUG-234: LESSON IN BASIC - NUMBER ONE

First lesson in BASIC covers: READ, DATA, LET, INPUT

IUG-235: MEMORY DRIVER

This program in Basic Assembler language requires BOSS/4B. This program allows the computer memory to be used as a peripheral (Device X'FE') as far as BOSS/4B is concerned. It can be forward spaced, backspaced (108 byte records), EOF, rewind, forward file, backward file. It is useful for program rearrangement, a one mag-tape system and, in general, for any object code operations.

PROGRAMS IUG-236 - 285 ARE WRITTEN IN DARTMOUTH TIME SHARING SYSTEM

BASIC AND ARE AVAILABLE IN LISTING FORM ONLY.

IUG-236: ANNUITY

How long before YOU go broke??? This program will figure out how long your money will last depending on the interest and withdrawal rates of your savings plan.

IUG-237: CALENDAR

A program which will print an entire calendar for any given year, print the calendar for any given months in any given year, or print the day of the week for any given date.

IUG-238: POPULA

Among the many applications of the compound interest formula is that of population projections. It is not entirely accurate in that it assumes a steady increase each year, but it is useful for showing roughly how an area will increase. This program will generate data for any number of years at any requested intervals.

IUG-239: FORECAST (FCST )

Analyzes data containing no trend component or a linear trend, with no seasonal pattern, or with a constant or multiplicative seasonal pattern. The user specifies the model to be fitted to the data. If the user specifies no trend, a simple average is used for forecasting. If a trend is specified, linear regression is used. The test section of the program uses only part of the data to initialize the model and then forecasts several periods ahead, so that the quality or fit of the model can be observed. Then the forecast section of the program uses all the data to initialize and forecasts values for several periods beyond.

IUG-240: FOURIER

Calculates and saves in a file the Fourier coefficients of any given function  $X(t)$ . This file can be used as input to RESPONSE (IUG-241). The user also has the options of printing out the coefficients and the frequency spectrum at the terminal.

IUG-241: RESPONSE

Computes the Fourier coefficients of the periodic response of a system to a given periodic input. The input function is specified in a terminal-format file consisting of Fourier coefficients in the format that is generated by the program FOURIER (IUG-240). The system is defined by its frequency response. To plot the output of this program, use PLOTSKI (IUG-253).

IUG-242: LESSEE

Uses the Bower-Williamson method of analysis to compare alternatives of leasing a piece of equipment outright. The method calculates two cost differences between owning and leasing. The first, the financial advantage, is simply the difference between the amount of debt capacity used up by the load and that used up by the discounted present value of all cash flow advantages including depreciation tax benefits.

IUG-243: LESSIM

Calculates the rate of return which the lessor receives for investing in an asset and then leasing it to someone else. Unlike LESSOR (IUG-244), this program recognizes that the rental payments from the lessee and the salvage value of the asset are uncertain. Using an estimate of the chance of default or discontinuance in any year and an estimate of the possible variation of actual salvage, the program simulates the experience of the lessor. The user specifies the number of trials. The output indicates the lessor's expected return, the possibility of loss, and the distribution of random outcomes.

IUG-244: LESSOR

Calculates the rate of return which the lessor receives for investing in an asset and then leasing it to someone else; i.e., the interest rate which discounts all of the net cash flows back to the initial investment the lessor must make. By comparing this after tax rate of return with the returns expected from alternative investments, the lessor can determine the desirability of the lease. This rate of return is calculated from the lessor's cash flows, which depend on lease receipts and tax payments.

IUG-245: MODEL

Creates an input file for OPTION (IUG-246) from data statements. The program contains a complete set of sample data.

IUG-246: OPTION

Similar to PROJECT (IUG-247), but allows the user much greater flexibility in the manipulation of relationships between accounts. The user provides five years of historical data and the program projects five more years. Historical regression and proportion coefficients may be computed for use in projections.

IUG-247: PROJECT

Projects financial statements of a company for five years into the future. Input consists of the most recent year's balances for 27 key accounts. Output is written into a file saved by the user. An additional file may be specified to receive the output in the format required to project the statements for five more years.

IUG-248: SPREAD

Uses the output from PROJECT (IUG-247) and prints one of five financial statements: balance sheet, income statement, funds flow statement, balance sheet items as a per cent of total assets, and accounts as per cent of sales.

IUG-249: REPORT

Computes and prints financial statements based on the account manipulations performed by OPTION (IUG-246). Statements available are the income statement, balance sheet, and cash flow statement.

IUG-250: RISK

A risk analysis program based on Hertz simulation model described in Jan.-Feb. 1964 "Harvard Business Review" article. The user may design his own model to project a set of cash flows. He may supply one value for each factor in his model or may supply a number of possible values for each factor and the per cent chance of the actual value lying between data pairs. RISK uses this information with a random number to calculate a value for a particular factor and repeats the process to calculate several sets of cash flows and rate of return and a net present value for each set of cash flows. The program prints histograms which show how many sets of flows have generated rates of return or net present values which fall within certain intervals.

IUG-251: GRAPH

Plots the graph of a function. The X-axis is drawn in, and if X=0 in the given range, the Y-axis is also drawn in. User specifies the function in a 'DEF' statement and the minimum and maximum of the range, the spacing on the X-axis, and any undefined points in 'DATA' statements.

IUG-252: PLOT-IT

A program to plot points in two or three dimensions, on any terminal or on the high speed printer (when run in Background). The user may use the full width of the paper on the printer. Scales are set automatically to include all the points to be plotted, but the user may alter the scales.

IUG-253: PLOTSKI

Graphs functions defined by their Fourier coefficients. PLOTSKI can be used in conjunction with RESPONSE (IUG-241) and FOURIER (IUG-240) to plot the output of a system defined by its frequency response. FOURIER produces FOURIER coefficients of a periodic (input) function defined in the multiple line DEF statements. These coefficients, together with the frequency response of the system, can be used in the program RESPONSE to compute the Fourier coefficients of the output function of the system. Several sets of coefficients may be stored in a file, and the program allows the user to plot them separately. Sample data file for demonstration (PLOTSKID) is contained in PLOTSKI.

IUG-254: LEMT

Lunar landing simulation. The user is the pilot of a lunar module trying to land his craft on the surface of the moon. The user inputs the amount of fuel to be burned in each second of his descent. Display consists of graphs for altitude, vertical velocity, and amount of fuel remaining.

IUG-255: TEK10

A package of subprograms for obtaining graphical output on the TEKTRONIX T4010 CRT terminal.

IUG-256: COMMON

A package of subprograms for obtaining graphical output on any of the following devices: TEKTRONIX T4002, T4010, TELETYPE or other printer, Hewlett Packard 7200.

IUG-257: MAKE-BUY

A program designed to help a manufacturer decide whether to buy a certain component for his product, or make it in his own plant. The cost of buying the component is compared with the discounted stream of cash flows that would result if the necessary investments were made to produce the component.

IUG-258: PROFITS

The user enters the sales, purchases, and beginning and ending inventories for each department of a firm. The program computes and prints information about inventory turnover and profitability for each department.

IUG-259: CPM

Analyzes a PERT-time network (for product planning). For each activity in the network the program determines the mean completion time, the earliest expected completion time, the variance associated with the completion time, the primary slack, and the secondary slack.

IUG-260: LAYOUT

Uses the Vollamnn-Ruml plant layout model to determine the optimum layout of a plant with several departments. The departments are assumed to be rectangular and laid out in near rows. The user inputs a flow matrix for exchanges between departments, cost weightings, and starting solutions. One or more departments may be held in a fixed location.

IUG-261: DIFEQ

Solves a group of N First-order differential equations of the form:  
$$DX(I) = G(I) = 1,2,3,\dots,N,$$
when the initial conditions are known, by the fourth order Runge Kutta method.

IUG-262: INTERP

The user supplies a list of arguments and their corresponding function values, and the program interpolates the function value Y at a given point X, using the Aitken-Lagrange interpolation for a single valued function.



IUG-263: INTGRT

Computes the integral of any function, over any interval, using Simpson's rule, a technique which breaks up a curve into pieces and approximates each section with a parabola.

IUG-264: QUADROOT

Computes roots, both imaginary and real, of any quadratic equation (an equation of the form  $aX^2 + bX + c = 0$ ) using the quadratic formula, given values for A, B, and C. This program includes both a subprogram and a driver program.

IUG-265: RUNGE-2

Gives an approximate solution to the second order differential equation:

$$P' = F(X, Y, P)$$

$$Y_0 = Y(X_0)$$

$$P_0 = P(X_0)$$

using the Runge Kutta method (second order accuracy). This program includes both a subprogram and a driver program.

IUG-266: SIMPLEX

Uses a simple version of the Simplex method to solve small linear programming problems (limited to 60 activities, 30 constraints). This method is G. Hadley's two phase method described in his "Linear Programming", pp 149-158.

IUG-267: ANOVAR

Provides complete factorial analysis of variance for up to 14 factors. The analysis is performed by the use of three special operators. The analysis of most other designs can be derived by reducing them first to the factorial designs and then pooling certain components of the variance table.

IUG-268: LINFIT

Computes best linear fit for a set of up to 20 independent variables to a dependent variable. (The program may be easily modified for more than 20 variables). The program also gives the correlations between the independent and dependent variables.

IUG-269: MULTREG

Computes one or more multiple linear regressions on a batch of data. 211 is maximum number of data sets; 17 is maximum number of variables. Output is more elaborate (and longer) than that of STEPREG (IUG-271). User may specify whether or not variance-covariance matrix and/or residuals are to be printed out.

IUG-270: SIMPREG

Performs simple linear regression analysis on N sets of paired observations (X, Y), where X is the independent variable and Y is the dependent variable. The output includes the least squares estimates of the slope and intercept of the regression equation  $Y = A + B \cdot X$ , the analysis of variance table with the appropriate F value, the coefficient of determination, and optional output of predicted values and residuals.

IUG-271: STEPREG

Performs a stepwise multiple regression for up to 20 variables (can be expanded). The stepwise method allows the selection of the best subsets of the independent variables for 1, 2, ..., N-1 regressors (where the total number of variables is N). The regressors are chosen in order to maximize the partial correlation coefficient (or T-value) of the newly entered variable at each step. The final solution comes close to being the optimum combination of independent variables for maximizing the multiple correlation coefficient.

IUG-272: RUNOFF

A publications formatting routine for the printing of textual files on the Dartmouth Time-Sharing System (DTSS). Control words, or commands, inserted throughout the text determine the output format, including such items as page size, line width, titles, page numbering, and paragraphing.

IUG-273: STENO

A program for composing and automatically typing individual business letters. The program is simple enough to invite use for one-of-a-kind letters but general enough to handle form letters for a large mailing list. The program allows the use of variable text appropriate to each addressee.

IUG-274: STENOADD

This program (which can either be chained to by STENO (IUG-273) or run independently) is used to print addresses on envelopes or labels.

IUG-275: TEXTSUB

A BASIC "Library" file of subprograms for handling manipulation of strings and text. Subprograms included are:

- UCASE - changes lowercase letter to uppercase
- YES-NO - inputs a string and checks to make sure the user answered "Yes" or "No".
- NSPACE - deletes all spaces from a string
- NCONTROL- deletes all control (non-printing) characters from a string.

IUG-276: VAL

This a BASIC subprogram which can be used in place of the function VAL (A\$) to convert a string into the number which is represented by that string.

IUG-277: DRIVE

Provides a way for a BASIC program to control terminal modes.

IUG-278: PAYROLL SYSTEM

Designed to computerize a town's payroll records. The main program PAYROLL enables the user to calculate and compile payroll summaries, calculate and write payroll checks, and keep accurate records for all changes regarding employees. The program also uses 8 subprograms which are appended to it, and the user must create several (empty) data files before running any of the programs.

IUG-279: GROUP

Uses grouped data (up to 20 groups) to calculate various statistical measures; mean, median, variance, standard deviation, skewness, and coefficient of variability. Data are input during the program run.

IUG-280: SAMPSTAT

Computes the minimum, maximum, range, sum, mean, sum of squares, variation for one or more groups of data.

IUG-281: STATSUB

A BASIC Library of statistical subprograms. Subprograms included are:

F-PROB	Computes probability of F variable
CHI-PROB	Computes probability of CHISQ variable
T-PROB	Computes probability of T variable
NOR-PROB	Computes probability of Normal variable
BIN-PROB	Computes probability of Binomial variable
NOR-DEV	Generates a random normal deviate
POI-PROB	Computes probability of Poisson variable

IUG-282: FLOW

A flowcharting program which will print out a diagram of a BASIC program, including lines indicating all possible transfers of control. References all GOTO, GOSUB, IF...THEN, ON...GOTO, and FOR...NEXT instructions. May be RUN at a terminal or in BACKGROUN; user may specify a specific block of lines to be flowcharted.

IUG-283: COMMENT

Aligns programming comments in 'BASIC' programs. User may specify alignment column, add comments or blank lines, or replace existing comments. COMMENT also neatens the user's program by indenting the body of FOR-NEXT loops, subprograms, and function definitions.

IUG-284: MAILSORT

A program for maintaining a terminal-format file of mailing addresses. Allows you to delete, change, or insert individual addresses.

IUG-285: MAILLIST

Allows user to print addresses on mailing labels or envelopes on most terminals, or on a high-speed peripheral device; normally used in conjunction with MAILSORT (IUG-284).

IUG-286: DSCGEN - GENERAL LOADER FOR 2.5MB DISC

This program is a modified version of the standard General Loader 06-025R03. The DSCGEN loader retains all the characteristics of the standard GEN LDR and in addition, the message "ERROR AT DISC DRIVER" reports all errors that may occur in the disc driver that replaces the other peripheral drivers. This one message is typed on the teletype in response to all disc related errors. such as invalid sector number, device unavailable, write protect, and others.

IUG-287: OS AIDS/RTOS

OS AIDS/RTOS is a modification of Interdata's OS AIDS that can be established as an RTOS task and that does all I/O to the keyboard device with SVC 1 instructions.

IUG-288: TAPTRN (TAPE TRANSLATOR)

Submitted by: Robert Rosen, Harry Diamond Laboratories, Washington D.C.  
Minimum system requirements: Model 70, 80, 85, TTY, HSPTR/P  
Software Requirement: Loader, Extended FORTRAN IV and Run Time Library

This program converts ASCII paper tapes from one format to another. Specific characters can be deleted, specific trailing characters can be deleted (e.g. trailing blanks), special end of record sequences can be added (e.g. replace X-OFF by CR, LF, and 6 NULLS) and any parity can be generated. The user can control the speed of the HSPTR (useful when reading roll paper tape on fan-fold reader).

IUG-289: VIRTUAL DATA MEMORY

Submitted by: Robert Rosen, Harry Diamond Laboratories, Washington D.C.  
Minimum system requirements: Model 70, 80, 85, TTY, Disc. DOS

This routine enables the user to manage a virtual data memory of 64K bytes regardless of the size of real memory. The user supplies the routine with the virtual address (0 LOC 65535) and the routine returns the real address of the item, doing any paging necessary. The routine uses a modified least-active-page replacement algorithm. The user may vary the page size, the number of pages kept in memory, and the paging unit.

IUG-290: INTERACTIVE FORTRAN W/TRAP

This program includes the multiply/divide Trap routine for machines that do not contain the high speed option. This version is the only one available for users with only 8KB memory and no high speed option. Since the Trap arithmetic operations require more execution time than hardware instructions, this version of Fortran runs slower than the other versions. Also, due to the size of the Trap routine, the working space is reduced considerably. Therefore, this version of Fortran requires more than 8KB of memory to be extensively useful, and contains several user restrictions.

IUG-291: INTERACTIVE FORTRAN/HARDWARE FLOATING POINT

This program uses hardware floating-point instructions, and therefore requires the Floating Point Instruction set.

IUG-292: INTERACTIVE FORTRAN/SOFTWARE FLOATING POINT

Performs its calculations by software floating-point computations, and does not require, nor can it use, hardware floating point instructions.

IUG-293: OS INTERACTIVE FORTRAN

This program performs its calculations using either hardware or software floating-point computation subroutines. However, this method used by the system depends on the user's choice of the starting location.

IUG-294: DOUBLE PRECISION INTEGER DIVISION (DBLDVD) John Jones, Interdata Inc.

Requires: New Series Processor, 16KB Memory, OS, Library Loader  
Source Language: Assembler

This routine will divide a 64-bit signed dividend by a 32-bit signed divisor and yield a 32-bit signed quotient and a 32-bit signed remainder.

IUG-295: DOUBLE PRECISION INTEGER CONVERSION ROUTINE (DBLINT) J. Jones, Interdata Inc.

System requirements: As in IUG-294

This routine converts a double precision integer into its decimal equivalent.

IUG-296: DOUBLE PRECISION INTEGER MULTIPLICATION (DBLMLT) J. Jones, Interdata, Inc.

System requirements as in IUG-294

This routine multiplies two signed 32-bit integers and returns a signed 64-bit integer.

IUG-297: INTERVAL TIMING (INTTIM)

System requirements as in IUG-294.

This program may be used to provide the execution time between two points of a calling program.

This program requires three additional subroutines: DBLMT (IUG-296, DBLINT (IUG-295), and DBLDVD (IUG-294), to perform double precision arithmetic and conversion.

IUG-298: STAR TREK II Author: E. Shirdlu, Omnimaven Consultants

Star Trek II is based on the game SPACWR from David Ahl's 101 BASIC Computer Games. The II designation indicates considerable enhancements due to the author and many of his colleagues. Most of the names of the players and equipment are based on the TV show Star Trek, but no other similarity is intended.

IUG-298A: STAR TREK III Author: C. Archer, Interdata, Inc.

Yet another version in BASIC of the popular game Star Trek in which you, the captain of the spaceship must rid the galaxy of the deadly Klingon menace.

IUG-299: HUNT THE WUMPUS Author: E. Shirdlu, Omnimaven Consultants

This game written in BASIC is a hunt for the cave-dweller Wumpus. Wumpus occupies 20 rooms, each room has 3 tunnels leading to other rooms. The object of the game is to hit the Wumpus in spite of such hazard as bottomless pits and superbats.

IUG-300: CLUB-DOS Author: Robert Rosen, Harry Diamond Labs, Washington DC

This program is a modified version of OS CLUB, the on-line interactive hexadecimal debug program. The program runs on any Interdata 70, 80, 85, 7/16 etc. which supports an operating system and the STM instruction.

IUG-301: OS CLUE 70 (OS CLUB WITH TRACE) Author: H. Lyster, National Research Council of Canada.

Minimum requirements: Model 5 or 70, TTY, BOSS, 5B or equivalent

Language: Interdata OS Assembler

Available material: Source paper tape, Relocatable Paper Tape, Assembly Listing

Occupies: X'108C' Bytes

OS CLUE 70 is an extension of the Interdata Hexadecimal DEBUG Program 'OS CLUB' (03-032). The operation of OS CLUE is identical with that of OS CLUB as described in the CLUB Manual, except for the addition of directives for selective program trace and sequential program input and the substitution of the letter 'U' for the letter 'O' as the directive which outputs labelled binary data in loader format.

IUG-302: IFFUNC/BOSS 4B SCOPE DRIVER Author: Marilyn Sealey, University of Iowa, Speech & Hearing Center.

Minimum requirements: Model 4, 16KB Core Memory, Floating Point, Tektronix Scope 603, BOSS 4A or 4B, Interactive Fortran

This program is a modification of IUG-123: READ/WRITE FUNCTIONS and adds functions for plotting to a scope. The scope driver can be patched into BOSS, 4B.

IUG-303: VARLIST  
LANGUAGE: BASIC  
PROCESSOR: 16-bit

Author: B. Blicht  
General Electric Company  
St. Petersburg, Florida

This program reads and stores all variables (\$ and numeric) and prints an alphabetical list of the variables. A useful program for adding another variable when "What is left" is not known.

IUG-304: XQDOS  
LANGUAGE: FORTRAN IV  
PROCESSOR: 16-bit

Author: Dr. F.M. Kruger  
National Center for Deaf-Blind  
New Hyde Park, N.Y.

This is a serially re-useable subroutine to execute a DOS command from a FORTRAN program.

IUG-305: LOADER FORMAT  
ANALYZER  
LANGUAGE: FORTRAN V, Level 1

Author: R.H. Waterman  
INTERDATA, Inc.  
Oceanport, N.J.

System Requirements: INTERDATA 16-bit processor, a binary input device and line printer, an Operating System, Library Loader and FORTRAN V, Level 1 Run Time Library, 16KB of memory plus the amount needed for the operating system.

The Loader Format Analyzer reads standard 16-bit format records and provides a printout for each record showing:

- a. sequence number
- b. checksum value
- c. loader items
- d. data items

IUG-306: GAMES 2  
LANGUAGE: BASIC  
PROCESSOR: 16-bit

Author: Dr. F.M. Kruger  
National Center for Deaf-  
Blind Youths & Adults  
New Hyde Park, N.Y.

This program plays the game of "Animal, Vegetable, or Mineral?"



IUG-307: INTEL 8008 & 8080  
CROSS ASSEMBLER  
LANGUAGE: FORTRAN & ASSEMBLY  
PROCESSOR: 7/16  
SOFTWARE: DOS

Author: Robert Old  
EXTEL Corporation  
Northbrook, IL.

The Intel Cross Assembler runs on an Interdata 7/16 with the DOS operating system. The assembler is capable of assembling Intel 8008 and 8080 code.

The software comprises some 26 source modules, including a FORTRAN main body routine, 22 subroutines, 2 test routines and an optimized loader for the Intel processor.

The program is available on a user-supplied 9-track mag tape in source format.

IUG-308: COMPAR  
LANGUAGE: FORTRAN  
PROCESSOR: MODEL 80

Author: Bruce Williams  
National Acoustic Laboratory  
New South Wales, Australia

This program reads in two direct physical files from LU-1 and LU-2. A comparison is made between each group of sixteen halfwords in the file and if a difference occurs then the sixteen halfwords from the LU-2 file are printed out in LU-3 with a character translation. This program is particularly useful in comparing a program before and after runtime. Thus providing an aid for debugging the program.

IUG-309: DSKCMP  
LANGUAGE: RTRAN  
PROCESSOR: Model 80

Author: Bruce Williams  
National Acoustic Laboratory  
New South Wales, Australia

This program is to be run on a two disc drive system. It will copy all files on pack C6 onto pack D6 and will carry out a compression process for cleaning up the packs at the same time.

One advantage of the system is that it will provide a consistency between the busy flags and the file directory on the disc which may have become inconsistent particularly on a two disc system environment.

IUG-310: MAPSRT  
LANGUAGE: FORTRAN  
PROCESSOR: Model 80

Author: Bruce Williams  
National Acoustic Laboratory  
New South Wales, Australia

This program expects the LU-4 disc file to be positioned after the map output from the library loader. The program reorders the symbols into an alphabetic order and then into a numeric order, and outputs to LU-3.

The program outputs to LU-2 disc file, source of all the defined symbols in the map plus their entry points. This file can then be used as a dummy entry module.

IUG-311: OLINTE  
LANGUAGE: FORTRAN  
PROCESSOR: Model 80

Author: Bruce Williams  
National Acoustic Laboratory  
New South Wales, Australia

OLINTE will analyze the library and produce a listing containing a line for each record in the library. It will output for each record the sequence number of the record and will indicate using symbols any labels, entry points, external or common references at end of module.

IUG-312: SCOPYS  
LANGUAGE: FORTRAN  
PROCESSOR: Model 80

Author: Bruce Williams  
National Acoustic Laboratory  
New South Wales, Australia

This program inputs from disc file, source statements including features of "SCOPYS" and will output to disc file the version acceptable to the fortran or assembler compilers.

IUG-313: SAVMEM  
LANGUAGE: FORTRAN  
PROCESSOR: Model 80

Author: Bruce Williams  
National Acoustic Laboratory  
New South Wales, Australia

This program will read the current CONTAB table from DOS and using this table, certain parts of memory will be written on LU-E disc file. In particular, this program is used for saving a program in memory on disc. If the same program is saved after running, then a comparison can be made between the two programs and any changed areas can be used in debugging. IUG-308 COMPAR exists for doing the comparison.

IUG-314: UPDATO  
LANGUAGE: FORTRAN  
PROCESSOR: Model 80

Author: Bruce Williams  
National Acoustic Laboratory  
New South Wales, Australia

The program expects LU-2 to be positioned after the new module. It also expects the library to be terminated by label ENDVOL. It will search the library for the module and if it is not found on the library it will replace the label ENDVOL with the new module on LU-2 and then add label ENDVOL on the end of the new module which becomes the end of the library.

IUG-315: GTCAL  
LANGUAGE: FORTRAN  
PROCESSOR: 16-bit or 32-bit

Authors: Cheryl Allen and Jim Farmer  
Georgia Institute of Technology

This is an Interdata CAL cross assembler to run on a UNIVAC 1108. There are two main programs in this package, one to build the hash-coded op-code tables for the assembler (TGEN) and the main driver for the assembler itself (GTCAL). Unless the assembler is to be run on a machine other than the UNIVAC 1108, new op-code tables will not have to be generated as they are included in the assembler itself. The programs were written in FORTRAN to make the cross-assembler as machine independent as possible. Machine dependent routines for input and output and character manipulation are flagged as such both in the documentation and in the program listings. The documentation for each program is in three parts: a diagram of the general program flow, a description of the common areas and data structures, and a short description of each subroutine.

Available on user-supplied 9-Track magnetic tape.

IUG-316: DOSCMD (ICMD)  
LANGUAGE: ASSEMBLER  
PROCESSOR: MODEL 70, 12KB

Author: David J. Fylstra  
Stanford Research Institute  
Menlo Park, Ca.

DOSCMD is a FORTRAN-callable subroutine which passes DOS system commands into the system via an SVC-4 supervisor call, returning the mainline FORTRAN program.

ICMD is an array of any length and any type, such that the DOS command is stored in the array as a string of 70-bit ASCII characters stored one per byte, starting with the leftmost byte of ICMD(1). The string can be stored in ICMD with a DATA statement, or entered with a READ statement. The string must be terminated by a blank or carriage return. Illegal command syntax or unexecutable statements will cause an unconditional return to DOS, as will the execution of a TRANSFER or SVC 2 PAUSE.

IUG-317: DOS FOR MOD 4  
LANGUAGE: OS ASSEMBLER  
PROCESSOR: MODEL 4

Author: M. Sealey and T. Windman  
University of Iowa  
Wendell Johnson Speech &  
Hearing Center

This program is Interdata's Disc Operating System modified to run on a Model 4, and can do everything that a Model 70 can do. (See DOS Reference Manual 29-293R01).

IUG-318: GEPAC30 PATCH MODULE  
TO INTERDATA CROSS  
REFERENCE ASSEMBLER

Author: W.D. Cave  
General Electric Company  
Nela Park, Ohio

The GEPAC-30 instruction set is compatible with Interdata assemblers with the exception of the extended short branch instructions. A solution for this deviation is provided in the form of a patch module program to run with Interchange program IUG-215 Cross Reference Assembler, where the special GEPAC-30 mnemonics are defined using the "user defined op-codes" feature of the assembler, and operand expressions are properly evaluated by the patch module.

IUG-319: SNAP2  
LANGUAGE: OS ASSEMBLER  
PROCESSOR: MODEL 70

Author: K. R. Morin  
MacDonald Dettwiler & Assocs. Ltd.  
2182 West 12th Avenue  
Vancouver, B. C., Canada

SNAP2 is a simple core save and restore program for Interdata Model 70 with Intertape cassette. The user specifies the first, last and start address of the core area to be written by setting up core locations. SNAP will write a single record containing a bootstrap plus the core image specified. The core image can then be read into core using the standard 50 sequence (or LSU program 97-000001).

Features: dynamic display, checksum, semi-automatic error recovery, inhibit auto start, inhibit tape stop on error, read without destroying locations X'80 -X'CF', write routine callable as subroutine via EXTRN, optional automatic rewind after reading, multiple-record capability.

Size: 2000 bytes.

IUG-320: SSP - SCIENTIFIC SUBROUTINE PACKAGE

Language: FORTRAN

Contributed by: Roy Stehle, Stanford Research Institute,  
Menlo Park, California

Medium: This package is available on a user-supplied 9-track 800 bpi magnetic tape, accompanied by a library description organized and segmented by program function. The subroutines are in relocatable object format with individual labels.

The Scientific Subroutine Package is formulated for greatest ease-of-use with maximum flexibility. A total of 356 subroutines provide a broad base of computational routines aiding the user to build his own program. The user furnishes, as part of his program, whatever input/output and other operations necessary for total solution of his problem.

All routines are accessible by simple FORTRAN CALL statements, and are treated as standard FORTRAN subroutines; all data is transmitted via the arguments in the CALL statement and no outlines cause input or output directly, resulting in a versatile device independent service for the user.

Program is available on two user-supplied 2400' mag tapes.

IUG-321: INTERACTIVE GENERAL LOADER

Language: ASSEMBLY

Contributed by: W. Lowzik, American Business Computers,  
220 Montgomery Street, San Francisco, CA

Minimum system requirements: Model 4, 8KB, TTY or CRT on PASLA,  
2.5 MB Disk.

Media: Source available on user-supplied magnetic tape.  
Object available on paper tape

This version of the General Loader supports the 2.5 to 10 megabyte disk. It will run on any Interdata CPU on a PASLA or TTY on TTY interface (see low core).

Object programs may be loaded and linked from a DOS R03 physical disk file into core (must be attribute X'20') or 256 byte record length. There is no de-blocking done with a 256 byte record. It is only necessary to know where the first record in the program resides, i.e. sequence number X'FFFF'. The operating system supporting the disk may be DOS, BOSS, RTOS, OS-16MT, etc. however, DOS R03 will supply the required cylinder in decimal, where the program resides.

IUG-322: INTERCHANGE GAMES PACKAGE

Pacakage consists of: IUG-299: Hunt the Wumpus, IUG-298: Star Trek II, IUG-298A: Star Trek III, IUG-178: Golf Game, IUG-177: Yahtzee, and CASINO which includes, Craps, Roulette, One-arm Banit, Black Jack, Red Dog, Draw Poker.

Available on a user-supplied magnetic tape or cassette

IUG-323: MOTOROLA M6800/INTERDATA Cross Assembler

G.A. Thomsen, Motorola, Schaumburg, IL. 60172

Language: Interdata Assembler

This assembler runs on any Interdata 16-bit configuration capable of supporting an operating system. The program requires a scratch device to perform automatic two-pass assemblies. However, the source statments may be re-read from the original input device on pass two.

The assembler accepts source code for the M6800 microprocessor in standard Motorola format. The operand data identifiers, H, O, and B are not permitted. However, the leading identifiers \$, % and @ are permitted, allowing compatibility with existing Motorola source programs. Additionally, "conditional" assembly is permitted by employing "IF" statements of the type used in the Interdata OS Assembler.

The output consists of an object program loadable by both EXORCISOR<sup>TM</sup> and MIKBUG<sup>TM</sup> resident loaders. A listing and unsorted symbol table are also produced.

The assembler source listing is completely commented. Operating instructions, logical unit assignments, and notes pertinent to the assembler are included as the first page of the listing.

This program is available to Interchange members on a user-supplied magnetic tape or cassette.

IUG-324: PRINCIPLE, INTEREST RATE PROGRAM (31 STATEMENTS)

C. Bowman, USAFAS, Fort Sill, Oklahoma

Language: FORTRAN IV Processor: 16-bit or 32-bit

IUG-325: AMORTIZATION PROGRAM (68 STATEMENTS)

C. Bowman, USAFAS, Fort Sill, OK.

Language: FORTRAN IV Processor: 16-bit or 32-bit

IUG-326: MATRIX ALGEBRA, LINEAR EQUATIONS (130 STATEMENTS)

C. Bowman, USAFAS, Fort Sill, OK.

Language: FORTRAN IV Processor: 16-bit or 32-bit

IUG-327: TREND ANALYSIS PROGRAM (234 STATEMENTS )

C. Bowman, USAFAS, Fort Sill, OK

Language: FORTRAN IV Processor: 16-bit or 32-bit

This program consists of a main program and two subroutines

IUG-328: GRAPH ROUTINE (TEKTRONIX 4010-1)

C. Bowman, USAFAS, Fort Sill, OK

Language: FORTRAN IV Processor: 16-bit or 32-bit

IUG-329: SINE AND COSINE WAVES 1-720 DEGREES (TEKTRONIX 4010)

C. Bowman, USAFAS, Fort Sill, OK

Language: FORTRAN IV Processor: 16-bit or 32-bit

This programs draws on X and Y axes.

IUG-330: PLTBIO (BIORHYTHM CHART PLOTTER)

S. Baker, Naval Air Rework Facility, San Diego, CA.

Language: FORTRAN V

This program will plot your biorhythm chart on a monthly basis, thus enabling the user to plan future activities to take the best advantage of Physical, Emotional and Intellectual capabilities at any given time.

IUG-331: DISCUPT UTILITY (DISC PATCH)

R. P. Jung, Interdata GMBH, Munich, Germany

Language: FORTRAN V Level 1      Processor: 16-bit

Operating System: OS/16MT2

This program provides the capability to read a random disc file, one sector at a time (256 bytes), modify data in hexadecimal form, and create a new file and write onto it.

IUG-332: MULTI-USER INTERACTIVE FORTRAN SYSTEM

Keith Hobson, Interdata Ltd., Uxbridge, England.

The Multi-User Interactive Fortran System is a computational system designed for use in schools, offices and laboratories. The system uses a language which is a subset of FORTRAN, augmented to make it interactive. It has the power of such other interactive languages as 'BASIC', and possesses the advantage that its syntax is that of FORTRAN. This means that the system provides an excellent method for learning Fortan programming.

The system is physically operated from teletype terminals. These are connected by means of telephone lines to a central computer which provides the computational power. The users dial up the system whenever they wish to use it.

IUG-333: PUNCH

Jeff Sumberg, Interdata, Inc., Oceanport, N.J.

System requirements: 16-bit CPU, 8KB Memory, TTY, HSPTR/P  
Language: Basic Assembler

This program permits the user to punch characters from A-Z or 0-9 at the start of paper tape for the purpose of labelling, etc.

IUG-334: SOURCE MAINTENANCE UTILITY TASK  
J.N. Walker, H.M. Huber Corporation, Edison, N.J.

System requirements: 7/16 HSALU or other 16-Bit CFU with RTOS capability,  
disc or drum, I/O Device.  
Memory: 7KB above OS (not including memory requirements for editing, compiling  
or assembling)

This utility task creates and maintains directoried card image ASCII source  
file libraries in a direct access, mass storage environment. The program  
is designed as an aid to program development, in that it can be used to  
create, edit or compile source code interactively or in batch, with a  
minimum number of operations required on the part of the user.

IUG-335: UT200 EMULATOR FOR DEDICATED 7/16 using modified DOS plus a  
user program.

IUG-336: UT200 EMULATOR for 7/32 under OS 32/MT with ITAM

IUG-337: DCT-1000 MULTIPLEXOR EMULATOR FOR UNIVAC 1100

IUG-338: GTCALRO2 - FORTRAN VERSION OF CAL designed to be machine  
independent. This version has machine dependent routines for UNIVAC 1100.

IUG-339: EXPANDED PRECISION ARITHMETIC PACKAGE - EPAP  
Chris Henrich, Interdata, Inc.

System requires: 32-bit processor, FORTRAN Compiler, CAL, TET

This package performs arithmetic operations to a high degree of precision:  
up to 98 hexadecimal digits of significance.  
The number of digits found in any particular calculation is adjustable by  
the user; generally, more digits require more computation time. The  
package has two parts: (a) a main program designed for use at a terminal:  
(b) the subroutines which perform the actual arithmetic.

IUG-340: OS/32MT UNIVERSAL CLOCK MODULE & UNIVERSAL LOGIC INTERFACE DRIVER  
Jeff Waxler, Interdata, Inc., Dall, Texas

System requires: Model 7/32; 2 or more Universal Clock Modules, 1 Line  
Frequency Derived Clock, 1 or more Universal Logic Interfaces.  
Language: INTERDATA Assembler

The Universal Clock Driver provides support of additional Universal Clock  
Modules or a Line Frequency Clock under OS/32 MT. The supported attributes  
include: Read, Write, Wait, Proceed, Unconditional Proceed and Halt I/O/  
Intervals from 1 us to 4.66 hours are supported by the PIC.  
The Universal Logic Interface Driver provides very basic support for the ULI  
(under OS/32MT).



IUG-341: 70X LOADER FORMAT ANALYZER

System Requires: OS, Library Loader, and FORTRAN V Level 1 Run Time Library  
Memory: 16KB Plus memory required for OS

Reads the output produced by the loader from LU2 and prints its own output on LU3. The output of the analyzer is intended to be an easily readable form of the bit output of the loader.

IUG-342: DISCREL

System requires: 32-bit CPU. Language: CAL  
This Rel Loader can be used to load an OS/32 module from a disc file.

IUG-343: DIRECT Language: CAL Requires: 32-bit CPU  
DIRECT is a utility program which transforms the output of DISPLAY FILE command into a useful CSS file.

IUG-344: PSEUDO-RANDOM NUMBER GENERATOR SUBROUTINE PACKAGE  
Guy Sotomayor, Dickinson College, Carlisle, Pa.

This package, which can be supplied on a user-provided magnetic tape, consists of a set of subroutines for sampling from the uniform, Gaussian and exponential distributions. These pseudo-random number generator routines are conversions to the Interdata 32-bit series systems from the original routines designed by Dr. Jo Ahrens, Nova Scotia Technical College, for IBM Systems. Mathematical proofs for all the algorithms used exist in the recent mathematical literature. The routines in this package are low in memory requirements, execute in less than 3/4 of a millisecond and generally represent the state of the art in the generation of pseudo-random numbers.

Minimum requirements: 32-bit CPU, CAL Assembler, FORTRAN V compiler (for the test programs), Line Printer (helpful since the documentation is embedded in the source code as a prolog.

IUG-345: OS/32 ASCEBC C. Archer, Interdata, Inc.  
This program converts 80-byte ASCII records into 80-byte EBCDIC records. The program uses table look-up to convert the 64 character set, as well as 25 control characters.

IUG-346: EBCASC  
As above except that the conversion is from EBCDIC records into ASCII.

IUG-347: 1130 COMMERCIAL SUBROUTINE PACKAGE (Available on a user-supplied 2400' 9-T magnetic tape.)

This package contains three files. First, the 32-bit object code file, ready to link into the program as is. Second, the source file for the subroutines. The last file contains the source code for all the test programs.

The source modules are completely documented, internally. However, for further information refer to the IBM 1130 Commercial Subroutine Programmers Guide H20-0241-3. As well as the 1130 CSP routines, several commonly used additions and a sort/merge procedure have been included.

Most of the subroutines are coded in standard FORTRAN IV. Most of the routines have been tested, used successfully, and have corresponding test programs in the test program file. The rest should be used with caution. The exception is EBCASV. This routine does not have test program but has been tested by verifying its operations through the use of OS/32 AIDS.

The routine TEXT was coded entirely in CAL Assembly code. TEXT has not been tested in the 16-bit mode. The LSDMF executive routine to the sort/merge procedure uses online assembly statements to accomplish dynamic memory management. The technique used will require some modification for systems other than OS/32ST.

IUG-348: CANONICAL CORRELATION ANALYSIS - CANONA

System requires: 32-bit cpu. Available on user supplied magnetic tape.

This program was designed to provide the research with the facilities necessary to study experimental results without many of the restrictions imposed by the more traditional analytic methods. A by-product of canonical correlation allows the user to study the affect of various groupings of variables from one set on the individual variables of the other set. It should be noted however that data to be analyzed by this program should fit the definition for the interval or ratio measurements scales. Data that include frequencies of responses to questions, answers, counts of occurrences, and readings from chemical and physical analytical instruments fall into this category.

IUG-349: NSTAT - NONPARAMETRIC STATISTICAL ANALYSIS & CROSS CLASSIFICATION (Available on a user-supplied magnetic tape)

This program was designed to provide the researcher with nonparametric procedures analogous to the more common parametric statistical procedures.

The facilities provided by this program for analyzing data are those procedures that involve the user of contingency or cross-classification frequency tables.

IUG-350: 7-TRACK MAGNETIC TAPE DRIVER

Requires: 32-bit cpu, OS/32MT

This driver is designed to operate on an Interdata 7/32 as part of OS/32MT. It may be included with the operating system at Sysgen time by specifying a device code of 240, and providing cpu with a file containing the DCB and the 7-Track driver. The driver supports both ASCII and BINARY I/O as binary.

IUG-351: DISCUPD UTILITY

R.P. Jung, Interdata GmBh, Munich, Germany

This program provides the capability to read a random disc file, one sector at a time (256 bytes), modify data in hexadecimal form, and create a new file and write to it.

System requires: 32-bit cpu, OS.32MT

Language: FORTRAN V, Level I

IUG-352: OS/16MT-2 SYSTEM TABLE BUILDER II

Greg Palmer, RCA, Camden, New Jersey

This program operates and runs exactly as the OS/16MT-1 System Table Builder program (03-079), with the following considerations:

1. The program can be executed under OS/16-MT2.
2. The source output file can be assembled under the CAL/16 (03-101) assembler.

The following restrictions/enhancements have been made:

1. A binary label, TNBLD, has been inserted.
2. A transfer address has been added ( $\emptyset R$ )
3. When executing under OS/16MT-2 the option COMP must be specified, and all optional logical unit assignments which are not used, must be assigned to the NULL device.
4. The DLIST pseudo-op has been replaced with comparable coding, since the DLIST statement is not supported by CAL.16 (this feature is transparent to the user).
5. The PREAMBLE statement must be specified to override the default preamble which would generate a "WIDTH 120" statement which is not supported by CAL/16.

IUG-353: FIND

David J. Fylstra, Stanford Research Institute,

This Fortran-callable subroutine searches a library file on logical unit LU for a labeled object module. It accomplishes the same function as a Library Loader FIND command.

System requires: Model 70, 16KB Memory, Fortran Compiler, Library Loader.  
Language: Assembler

IUG-354: FETCH

David J. Fylstra, Stanford Research Institute  
Menlo Park, CA

This subroutine allows the DOS users to fetch and load a labeled overlay segment from a library of labeled object modules. Other subroutine used is: FIND (IUG-353).

IUG-355: SCAN

David J. Fylstra, Stanford Research Institute  
Menlo Park, CA

This FORTRAN-callable subroutine is a general character scanner. The subroutine scans across character string LINE (packed one character per byte) from position START to position END, until a delimiter is found. The intervening string is returned in ITOKEN, and START is left positioned at the delimiting character; the delimiter is returned in the left byte DELCHR. LINE and ITOKEN may be any type of array receiving characters from FORTRAN READ or DATA statements; character positions are counted left-to-right starting with zero. START, END, and DELCHR are typically two-byte integers.

IUG-356: TIC-TAC-TOE

C. Archer, Interdata, Inc., Oceanport, N.J.

This version of the game TIC-TAC-TOE is written in BASIC.

System requires: TTY or CRT; BASIC; BOSS, OS/16MT2 or OS/32MT.

IUG-357: REAL-TO-ASCII & INTEGER-TO-ASCII FIXED FORMAT CONVERSION  
SUBROUTINES

(DOCUMENT ONLY AVAILABLE)

W.D. Cave, General Electric Company, Nela Park, Cleveland, OH

This conversion package is designed for real-time small memory system applications on Interdata 16-bit computers. The routines are compact and fast. All parameters are passed via registers, and the routines are completely re-entrant. They are fixed format as opposed to free-format in that the ASCII result is stored (right justified) in a user buffer of specified size.

IUG-358: CRIBBAGE

Charles V. Codling, Interdata Inc,  
Washington, D.C.

This Cribbage program plays you against the computer. A Random Number Generator is used to determine hands dealt and who has the first Crib. RAND.CAL is a Random Number Generator. An Equate statement is at the front to indicate FORTRAN V or VI calling sequences and should be set for the respective compiler used.

INPUT = Logical Unit 1      OUTPUT = Logical Unit 6

System requires: Model 6/16 - 8/32, 30KB Memory, KVDT or Carousel Device.  
Language: FORTRAN V Level I

IUG-359: SNAPSHOT (Dump subroutines)

Charles V. Codling, Interdata, Inc.  
Washington, D.C.

System requirements: 7/32, 8/32, .25KB Memory

Prints hex and ASCII equivalent data on LU-6 from the starting through ending addresses. These addresses are rounded to word boundaries in multiples of 8 words. The start address is rounded downward and the ending address is rounded upward, if required. Multiples of the same line are suppressed with an asterisk on the last line of the repeated lines.

IUG-360: MULTI-USER EXECUTIVE OPERATING SYSTEM FOR 16-BIT CPU

The BASIC Multi-User Executive is a dedicated operating system specifically tailored to provide a multi-terminal environment using the BASIC Interpreter (IUG-365). As few as two or as many as 32 terminals may be supported. Although designed specifically for use with the BASIC Interpreter, the Executive can support other re-entrant programs.

The modular construction of the Multi-User Executive allows each user to specify the type of terminal device to be used and the address of the terminal. A full range of I/O device drivers is supported, allowing user written drivers to be added or removed.

#### IUG-361: FIXED AND FLOATING POINT MATH PACKAGE

This package provides a variety of routines for 16-bit arithmetic operations. The 4 binary-to-decimal type conversions do not use multiply-divide op-codes. The various trigonometric routines do use multiply-divide op-codes. The random number generator routine produces pseudo-random binary sequences of 1 to 16 bits.

The package consists of:

- Binary to DEcimal Integer Conversion
- Binary to Decimal/Fractional Conversion
- Decimal to Binary Integer Conversion
- Decimal to Binary Fractional Conversion
- Square Root
- Log Base 2,e.10
- Exponential
- Sine/Cosine
- Arctangent
- Angle Conversion
- Multiply/Divide Trap
- Random Number Generator
- Fixed-Point Single Precision Conversion Routines
- Fixed Point Single Precision Arithmetic Routines.

#### IUG-362: TIDE (16-BIT TEXT EDITOR)

TIDE is an on-line, interactive text editing program that assists the user with creation, examination and manipulation of source program paper tapes. The user directs OS TIDE through keyboard commands writing the textual information within TIDE's text buffer until a hard copy of the text is obtained (listing/tape). TIDE assumes the teletype is interfaced to the processor as Device 02. TIDE runs on any INTERDATA 16-bit cpu. The program requires approximately 5K of core memory.

#### IUG-363: RTOS - REAL TIME OPERATING SYSTEM - 16-BIT MODEL 70/80

RTOS provides the user with the capability of interleaving the execution of a number of programs based on their established priorities. While still being responsive to the occurrence of real time events and their required processing, RTOS can perform a variety of background tasks.

#### IUG-364: RTEX - REAL TIME TELECOMMUNICATIONS EXECUTIVE

Under RTEX, program tasks (channel programs) issue requests to the system for operations on communication adapters with the Supervisor Call 15 instruction (SVC15). System servicing of communication calls under RTEX offers considerable latitude in the manner in which communication drivers may be designed and executed. This flexibility enables communication systems to be customized for a given application.

IUG-365: BASIC INTERPRETER

BASIC is a widely accepted, general purpose, interactive programming language designed to include extensive programming power along with simplicity of use. This BASIC Interpreter provides the BASIC language under any existing INTERDATA operating system and at R)@ and above for INTERDATA 16-bit and 32-bit processors.

INTERDATA BASIC is compatible for the most part, with BASIC written for other systems. Programs to be converted should require only minor modification.

The BASIC Interpreter is written as a re-entrant package and can provide multi-user capability and file handling features within the BASIC language itself when run under the multi-user capability under RTOS and OS/16MT on the 16-bit processors and under OS/32MT on the 32-bit processors.

BASIC requires a Model 5, 70, 7/17, 80, 7/32 or 8/32 processor or equivalent; or a Model 74 or a Basic 7/16 using an OS with floating point trap support. BASIC operates only with the previously named operating systems and may not be supported by any other OS which might be provided by INTERDATA.

IUG-366: FLOXOS - REAL TIME BASIC WITH FLOPPY DOS

W. Hall and P. Stein, National Bureau of Standards, Washington, D.C.

System requires: Basic 6/16, 32KB Memory, DOS, BASIC

Source Language: Assembler

Available on user-supplied magnetic tape. System consists of four files:

- File 1 - Calls source
- 2 - FLOXOS source
- 3 - XF2CPL SYSGEN
- 4 BASIC

All programs are inter-related and cannot be run separately without considerable rework.

IUG-367: OBDUMP - Object Dump

Eric Michelsen, Interdata, Inc.

System requires: Any Interdata processor, 3KB Memory, Binary object input, ASCII list device, OS/16MT2 or OS/32.

Source language - CPL

OBDUMP is a common mode program which lists 16 or 32 bit object code in an easily readable form. Binary object code is read from LU1 and a listing is sent to L03. The LU assignments must be made before starting OBDUMP

IUG-368: CAL-360/370 CROSS ASSEMBLER

Dr. A. Savitzski, Perkin-Elmer,  
Norwalk, Connecticut.

There are two main programs in this package, one to build the hash-coded op-code tables for the assembler (GCEN) and the main driver for the assembler itself (GTCAL). Unless the assembler is to be run on a machine other than an IBM 360/370, new op-code tables will not have to be generated as they are included in the assembler itself. The programs were written primarily in FORTRAN to make the cross-assembler as machine independent as possible. Machine dependent routines for input and output and character manipulation are flagged as such both in the documentation and in the program listings. The documentation for each program is in three parts; a diagram of the general program flow, a description of the common areas and data structures, and a short description of each subroutine.

IUG-369: MOVESTR

David J. Fylstra, Stanford Research Institute  
Menlo Park, CA

MOVSTR is a general purpose character-string manipulation routine. It Moves NCHARS characters from the string in array LINE 1 starting at position START1 to the string LINE2 starting at position START2. Character positions are numbered from left to right starting with zero. START1, START2, and NCHARS are two-byte integers, and line1 and LINE2 are character-strings packed one character per byte in any kind of array.  
System requires: Model 70, 16KB Memory, Assembler, FORTRAN compiler.

IUG-370: BFP - BASIC FORTRAN PLOTTER

Dr. K. Foerster, Institute for Aerodynamics & Gasdynamics,  
Stuttgart, Germany

System requirements: Model 70, TEKTRONIX 4006-1, FORTRAN IV  
Available material: Source listing, description

BFP is a short subprogram (9 FORTRAN IV statements) performing the following basic task of plotting: Given a polygon by the set of Cartesian coordinates of its initial, corner and end points. Draw the polygon on the CRT's screen. The coordinates must already be 2-byte integers in the range displayable by the CRT ( $0 \leq x \leq 1023$ ,  $0 \leq y \leq 780$ ); the number of points (being 49 in the listed version) can be extended easily.

```
SUBROUTINE BFP(G,N)
  INTEGER*2 GS,G(N)
  GS=29
  DO 1 I=1,N
1  G(I)=G(I)+G(I)/32*224+8288-MOD(I,2)*32
  WRITE(1u,2) GS,(G(I+1),G(I),I=1,N,2)
2  FORMAT(99A2)
  RETURN
END
```

IUG-371: FBCOPY - FIXED-BLOCK AND UNBLOCK COPY

J. Kendall & R. Fascenda, University of South Florida, Tampa, Fla.

System requires: 16-bit processor, 9KB Memory, Operating System to support SVCs, I/O dev.

FBCOPY is a useful program for blocking and deblocking. When blocking is performed, the input record must be less than the output record size. When de-blocking is desired, the output record must be less than the input record size. If this rule is not observed, then FBCOPY will respond with an error and ask for the record sizes again.

Example of deblocking an input record of 8,000 characters into 100 records of 80 characters.

```
RUN FBCOPY
FBCOPY /XX:XX:XX
INPUT RECORD
>8000
OUTPUT RECORD
>0080
EOJ XX:XX:XX
```

IUG-372: OS/32MT RUN ANALYZER

The OS/32MT Run Analyzer is a real-time collector of data as to system activity. The results of an analysis may be used to:

- Measure overhead and idle capacity under actual running conditions;
- Find bottlenecks and limiting factors in the operating system;
- Determine the potential payoff of proposed optimizations.

The analyzer portion of this program may also be patched to perform a similar analysis on tasks by:

BI \*  
VER 3A,423Ø  
MOD 3A,433Ø

IUG-373: SIMULATION PROGRAMS IN BASIC LEVEL I

S. W. Calvin, Department of the Army  
U.S. Dependents Schools

This is a package of educational programs for the study of biology, physics, and social studies.

IUG-374: SFS - SPOON FEEDER SYSTEM

The Univac 1110 Interdata 7/16 Interface has three functions. One, to send assembler images to the minicomputer for assembly; two, to record core images of the 7/16 for dumps or later re-loading; and three, to send the core images back to the minicomputer.

All of the communication is done through an asynchronous line.

The 1100 7/16 Spoon Feeder System consists of 5 interrelated programs:

- . SPOON/1100
- . DUMPER/1100
- . DUMPTER/716
- . PUNCHOBJECT
- . MODIFIED 7/16 ASSEMBLER



IUG-375: SORT

Dr. Samuel Raymond, Hospital of the  
University of Pennsylvania

The records are sorted in place without using any work area. The routine half-divides a given set of records into two halves so that all the records in the low half are less than the records in the high half. The mainline calls half repeatedly. Each time, the boundaries of the large half are sorted in stack and the small half is used. When the small half is eleven records or less, it is sorted by pairwise swapping and the previous half is examined.

IUG-376: CAL 360/370 CROSS ASSEMBLER

T. Whab, Bell Northern Research,  
Ottawa, Canada

This is an updated version of IUG-368. An 'EXEC' routine is supplied for the handling of the mainline FORTRAN code 'CALXASM'. This should be a ready to use 'CALXASM' cross assembler, with few if any changes required to run under IBM 360/370.

IUG-377: INPUT

Michael L. Campanella, Xerox Corporation,  
Rochester, New York

This is a blind I/O routine to input data without format specification.

IUG-378: SNAP

William Thomas, American Express Company  
Phoenix, Arizona

SNAP dump will allow the programmer to monitor general registers and storage as the application is executing. SNAP will print the contents of all general registers of the current set. It will also print the contents of specified storage.

#### IUG-379: DATA DIRECTOR EDITOR

The Data Director Editor is a powerful, multiuser editor that will aid in the development and modification of APT part program, assembly-language programs, and other text.

Some benefits of the editor are:

- . Most of the editor is reentrant, allowing several users to share it.
- . The user can utilize the editor as though the entire file being edited is in memory.
- . Editing takes place on a working file so that changes are not made to the original file until desired.
- . The editor offers a powerful command set where most commands have the same syntax.

#### IUG-380: BFP (BASIC FORTRAN PLOTTER)

(Listing and description only)  
Dr. K. Foerster, Institute of Aerodynamics &  
Gasynamics, Stuttgart, Germany

BFP is a short subprogram (9 FORTRAN IV statements) performing the following basic task for plotting on a Tektronix 4006-1: Given a polygon by the set of Cartesian coordinates of its initial, corner and end points. Draw the polygon on the CRTs screen.

The coordinates must already be 2 byte integers in the range displayable by the CRT ( $0 \leq x \leq 1023, \leq 1023, 0 \leq y \leq 780$ ); the number of points (being 49 in the listed version can be extended easily).

IUG-381: SOLO OPERATING SYSTEM INCLUDING SEQUENTIAL AND  
CONCURRENT PASCAL INTERPRETERS

Frank Kurka, Interdata, Inc.,  
Oceanport, New Jersey

Minimum system - operational OS/32MT2 et al, should include a disc and magnetic tape unit.

Sequential Pascal is an implementation language for utilities and languages.

Concurrent Pascal is an implementation language for operating systems and real time scheduling.

Solo is one operating system written in and implemented by Concurrent Pascal itself. Solo, in turn, supports both sequential and concurrent Pascal program development and operation by providing an environment for editing, file management and interpretation of Pascal programs, which is optimized for Pascal. In this sense, one operating system (Solo), runs under another (OS/32MTX), in a very transparent mode, thus providing a very different environment at the users terminal.

Included in the documentation are: bibliography, unpacking instructions, and certain manuals.

IUG-382: COBOL SQUEEZER

Kenneth Cohen, Interdata, Inc.,  
Detroit, Michigan

This program takes any Interdata COBOL/32 user-written source code and radically decreases the core requirements for the object module.

The COBOL SQUEEZER operates as a post-processor to the compiler and strips out certain redundant code and data from the CAL generated by the COBOL/32 compiler. It is invoked by a CSS call requiring only the name of the program to be squeezed, and a list device.

The program, which is fully tested, debugged and documented, is available on a user-supplied magnetic tape.

# ● INTERCHANGE

DATE \_\_\_\_\_ REQUEST NUMBER \_\_\_\_\_  
(For Office Use Only)

DELEGATE \_\_\_\_\_

ORGANIZATION \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY, STATE \_\_\_\_\_ ZIP CODE \_\_\_\_\_

### User Supplied Media

ITEM	IUG #	DESCRIPTION	DISC		MAG TAPE			DOCUMENTATION	SHIP DATE
			FLOPPY	2.5MB	10MB	CASSETTE	800 BPI		

In order to process this request as soon as possible, we ask your cooperation in requesting only four (4) programs per form.

\_\_\_\_\_  
Signature of Interchange User

REPLY      ITEM(s)  
 are not available at this time and will be shipped at a later date.

If there are any questions, please call. (201) 229-4040.

# INTERCHANGE

## SUBMITTAL FORM

Material submitted to INTERCHANGE for publication in the Newsletter or to the INTERCHANGE Program Library should be sent to INTERCHANGE, Perkin-Elmer Data Systems, 2 Crescent Place, Oceanport, New Jersey 07757.

Members of INTERCHANGE may contribute or request programs. Each program submitted should be accompanied by a program submittal form and should include source on an appropriate media as specified on the Program Order Form. Documentation should include a listing, operating instructions and a program abstract. As much other pertinent information as possible should also be included.

Wherever possible, the documentation should be typed or in an easily reproducible form. In most cases, the documentation will be reproduced as submitted without alteration.

Submittal forms may be obtained on request from the INTERCHANGE office.

TITLE \_\_\_\_\_

AUTHOR \_\_\_\_\_

ORGANIZATION \_\_\_\_\_

MINIMUM SYSTEM REQUIREMENTS TO USE PROGRAM OPERATIONALLY \_\_\_\_\_

PROCESSOR \_\_\_\_\_ MEMORY \_\_\_\_\_ KB OPTIONS \_\_\_\_\_

PERIPHERALS \_\_\_\_\_

SOFTWARE (other programs or subroutines needed) \_\_\_\_\_

SOURCE LANGUAGE \_\_\_\_\_

MATERIAL SUBMITTED:		<u>Format</u>
_____ SOURCE item(s)		
_____ OBJECT item(s)		
Occupies:		
Object Program Label		
START Address		
RESTART Address		
XFER		
Loads with		

TYPE OF LISTING SUBMITTED: Source \_\_\_ Assembly \_\_\_ Compiler \_\_\_

DOCUMENTATION: Description \_\_\_\_\_ Printout \_\_\_\_\_  
Flowcharts \_\_\_\_\_ Other \_\_\_\_\_

I, the undersigned, give full permission to INTERCHANGE to re-reproduce and distribute this program in full or in part, "free of charge" to all interested parties.

SIGNED \_\_\_\_\_



# INTERCHANGE

## PROGRAM REVIEW

The primary purpose of a program review is to determine whether or not the program operates as its author claims. The task of reviewing a program is an extremely variable one which cannot be specified in advance. The quality of the reviewing process, therefore, depends almost entirely on the diligence of the reviewer. As it is almost impossible to test every program that is submitted into the library in advance, we ask that you, the recipient of the program, test it out for us, complete this form, and return it to the INTERCHANGE office for evaluation. In this way, we can weed out ineffective programs and certify programs which do work efficiently.

Please check all statements made by the author about his program. When the program is in core, run it more than once to be sure that it has been properly initialized. Feel free to make comments about the program in your report. For example, if you know of a better or more general program for the same purpose, let us know.

PROGRAM REVIEWED: Number and Title \_\_\_\_\_

REVIEWED BY: Name \_\_\_\_\_

Organization \_\_\_\_\_

- |   | Yes                      | No                       |
|---|--------------------------|--------------------------|
| 1. Did the program compile?   | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Did the object program work?   | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Was it modified by you?  | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. Was the documentation adequate?  | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Are operating instructions adequate?   | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. Was the material included with the program sufficient for your use?  | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. Do you feel the program should remain in the INTERCHANGE Library?  | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. Do you have any suggestions for improving the usefulness of this program? (These comments or suggestions will be passed on to the author for his consideration.) |                          |                          |

---

---

---

---

---

---

---

---

Please return this form to:

INTERCHANGE  
Perkin-Elmer Data Systems  
2 Crescent Place  
Oceanport, New Jersey 07757



**INTERDATA®**

