

```

0000      0001 *
0000      0002 *
0000      0003 *
0000      0004 *
0000      0005 *
0000      0006 *
0000      0007 *
0000      0008 *
0000      0009 *
0000      0010 *
0000      0011 *
0000      0012 *
0000      0013 *
0000      0014 *
0000      0015 *
0000      0016 *
0000      0017 *
0000      0018 *
0000      0019 *
0000      0020 *
0000      0021 *
0000      0022 *
0000      0023 *
0000      0024 *
0000      0025 *
0000      0026 *
0000      0027 *
0000      0028 *
0000      0029 *
0000      0030 *
0000      0031 *
0000      0032 *
0000      0033 *
0000      0034 *
0000      0035 *
0000      0036 *
0000      0037 *
0000      0038 *
0000      0039 *
0000      0040 *
0000      0041 *
0000      0042 *
0000      0043 *
0000      0044 *
0000      0045 *
0000      0046 *
0000      0047 *
0000      0048 *
0000      0049 *
0000      0050 *
0000      0051 *
0000 31 90 17 0052 *
0003 3E 03 0053 *
0005 32 92 17 0054 *

```

PROCESSOR TECHNOLOGY BASIC-5

DEVELOPED IN CONJUNCTION WITH:

APPLIED COMPUTER TECHNOLOGY
1038 MERCED STREET
BERKELEY, CALIF 94707

ASSEMBLED USING THE ALS-8 PROGRAM DEVELOPMENT SYSTEM

*** COPYRIGHT 1976 ***

PROCESSOR TECHNOLOGY CORP.
6200 HOLLIS STREET
EMERYVILLE, CALIF.
94608 (415) 652-"8080"

<<<<<<<<-- ALL RIGHTS RESERVED -->>>>>>>

SYSTEM GLOBAL EQUATES

0034	FPSIZ	EQU	5	
0032	LINLEN	EQU	73	# CHRS IN LEGAL INPUT LINE
0033	FP123	EQU	FPSIZ-2	
0034	FPNIB	EQU	FP123*2	
0035	DIGIT	EQU	FPNIB/2	
0036	CR	EQU	15Q	CARRIAGE RETURN
0037	NULL	EQU	0	NULL CHARACTER VALUE
0038	LF	EQU	12Q	LINE FEED
0039	ESC	EQU	3Q	ESCAPE CHAR
0040	EOF	EQU	#	END OF FILE
0041	BELL	EQU	7	BELL CHARACTER
0042	STESIZ	EQU	2+FPSIZ	SYMBOL TABLE ELEMENT SIZE
0043	OPBASE	EQU	'('	
0044	FTYPE	EQU	1	CONTROL STACK FOR ENTRY TYPE
0045	FORSZ	EQU	FPSIZ*2+2+2+1	'FOR' CONTROL STACK ENTRY SIZE
0046	GTYPE	EQU	2	CONTROL STACK GOSUB ENTRY TYPE
0047	ETYPE	EQU	0	CONTROL STACK UNDERFLOW TYPE
0048	UMINU	EQU	64Q	UNARY MINUS

STARTUP BASIC SYSTEM

0052	START	LXI	SP,CMNDSP	
0053		MVI	A,3	
0054		STA	NULLCT	INITIALIZE NULL COUNT

0008 32 45 19	0055	STA	SPEED	
000B CD 68 0F	0056	CALL	INITS	
000E 33 6A 07	0057	LXI	D,FAS	FIRST ADDRESS MESSAGE
0011 24 53 19	0058	LXI	H,MEMTO+1	ONE LESS THAN LOWER BOUND
0014 CD 3B 10	0059	CALL	GINT	GET INTEGER GREATER THAN MEMTOP+1
0017 22 50 19	0060	SHLD	BOFA	START OF USER ASSIGNED MEMORY
001A 11 76 07	0061	LXI	D,LAS	
001D CD 3B 10	0062	CALL	GINT	HL HAS 1 LESS THAN LOWER BOUND AGAIN
0020 22 52 19	0063	SHLD	MEMTOP	END OF ASSIGNED MEMORY POINTER
0023 24 43 07	0064 ST0	LXI	H,PLS	"PROGRAM LOADED?" MESSAGE
0026 CD 23 0D	0065	CALL	PRNT	
0029 CD 55 0E	0066	CALL	INLINE	
002C 3A D3 18	0067	LDA	IBUF	
002F	0068 *			
002F	0069 *			OPTIONAL ENTRY POINT FOR TAPE OR DISK ROUTINES
002F	0070 *			
002F	0071 *			ALLOWS DIRECT PROGRAM INPUT FROM HIGH SPEED DEVICES
002F	0072 *			SEE OPERATING INSTRUCTIONS FOR PROPER IMPLEMENTATION.
002F	0073 *			
002F FE 4E	0074 STAR1	CPI	'N'	
0031 CA 51 00	0075	JZ	ST1	IF NO PROGRAM CLEAR AND INITIALIZE
0034 FE 59	0076	CPI	'Y'	
0036 C2 23 00	0077	JNZ	ST0	
0039 2A 50 19	0078	LHLD	BOFA	
003C 7E	0079 ST2	MOV	A,M	FIND END OF PROGRAM
003D FE 01	0080	CPI	EOF	
003F CA 48 00	0081	JZ	ST3	
0042 CD C1 01	0082	CALL	ADR	
0045 C3 3C 00	0083	JMP	ST2	
0048 22 4E 19	0084 ST3	SHLD	EOFA	
004B CD DF 01	0085	CALL	CCLEAR	
004E C3 54 00	0086	JMP	ST4	
0051 CD D7 01	0087 ST1	CALL	CSCR	
0054 3E 0C	0088 ST4	MVI	A,2*FPNIB	
0056 32 3A 19	0089	STA	INFES	
0059	0090 *			INITIALIZE RANDOM NUMBER
0059 11 CF 18	0091	LXI	D,FRAND	
005C 24 B1 07	0092	LXI	H,RANDS	
005F CD 9C 0C	0093	CALL	VCOPY	FRAND=RANDOM NUMBER SEED
0062	0094 *			
0062	0095 *			COMMAND PROCESSOR
0062	0096 *			
0062 CD 19 10	0097 CMND1	CALL	CRLF2	
0065 24 3D 07	0098	LXI	H,RDYS	PRINT READY MESSAGE
0068 CD 23 0D	0099	CALL	PRNT	
006B 3E 01	0100 CMNDR	MVI	A,1	SET DIRECT INPUT FLAG
006D 32 94 17	0101	STA	DIRF	
0070 31 90 17	0102	LXI	SP,CMNDSP	
0073 CD 1C 10	0103	CALL	CRLF	
0076 CD 55 0E	0104 CMND2	CALL	INLINE	GET INPUT LINE FROM OPERATOR
0079 CD 9F 06	0105	CALL	PP	PRE-PROCESSOR IT
007C DA 88 00	0106	JC	CMND3	
007F CD 10 01	0107	CALL	LINE	LINE NUMBER..GO EDIT
0082 CD DF 01	0108	CALL	CCLEAR	
0085 C3 76 00	0109	JMP	CMND2	
0088 CD 8E 00	0110 CMND3	CALL	CMND4	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

008B C3 6B 00	0101	JMP	CMNDR	
008E 21 D3 18	0112 CMND4	LXI	H,IBUF	POINT TO COMMAND OR STATEMENT
0091 22 95 17	0113	SHLD	TXA	
0094 CD 02 0D	0114	CALL	GC	
0097 E6 A0	0115	ANI	240Q	
0099 FE A0	0116	CPI	240Q	CHECK FOR COMMAND
009B 11 D2 07	0117	LXI	D,CMNDD	
009E CA 8E 02	0118	JZ	ISTAT	PROCESS COMMAND
00A1 CD 7F 02	0119	CALL	ISTAT	PROCESS STATEMENT (IF ALLOWED)
00A4 CD 0A 0D	0120	CALL	GCI	
00A7 FE 0D	0121	CPI	CR	
00A9 C8	0122	RZ		
00AA 01 53 42	0123 E1	LXI	B,'BS'	
00AD C3 C5 00	0124	JMP	ERROR	
00B0	0125 *	ERROR MESSAGE PRINT OUT		
00B0 01 41 42	0126 E3	LXI	B,'BA'	
00B3 C3 C5 00	0117	JMP	ERROR	
00B6 01 53 43	0128 E4	LXI	B,'CS'	
00B9 C3 C5 00	0129	JMP	ERROR	
00BC 01 42 4F	0130 E5	LXI	B,'OB'	
00BF C3 C5 00	0131	JMP	ERROR	
00C2 01 4D 44	0132 E6	LXI	B,'DM'	
00C5	0133 *			
00C5 C5	0134 ERROR	PUSH	B	
00C6 CD 1C 10	0135	CALL	CRLF	
00C9 C1	0136	POP	B	
00CA CD 95 0E	0137	CALL	CHOUT	
00CD 41	0138	MOV	B,C	
00CE CD 95 0E	0139	CALL	CHOUT	
00D1 21 54 07	0140	LXI	H,ERS	
00D4 CD 23 0D	0141 ERM1	CALL	PRNT	
00D7 3A 94 17	0142	LDA	DIRF	
00DA B7	0143	ORA	A	
00DB C2 62 00	0144	JNZ	CMND1	
00DE 21 5B 07	0145	LXI	H,INS	
00E1 CD 23 0D	0146	CALL	PRNT	
00E4	0147 *	FIND LINE NUMBER		
00E4 2A 50 19	0148	LHLD	BOFA	
00E7 44	0149 ERM2	MOV	B,H	
00E8 4D	0150	MOV	C,L	
00E9 5E	0151	MOV	E,M	
00EA 16 00	0152	MVI	D,0	
00EC 19	0153	DAD	D	
00ED EB	0154	XCHG		
00EE 21 95 17	0155	LXI	H,TXA	
00F1 CD 35 0D	0156	CALL	DCMP	
00F4 EB	0157	XCHG		
00F5 DA E7 00	0158	JC	ERM2	
00F8 03	0159	INX	B	
00F9 0A	0160	LDAX	B	
00FA 6F	0161	MOV	L,A	
00FB 03	0162	INX	B	
00FC 0A	0163	LDAX	B	
00FD 67	0164	MOV	H,A	
00FE 11 D3 18	0165	LXI	D,IBUF	USE IBUF TO ACCUMULATE THE LINE NUMBER STRING
0101 CD FA 0D	0166	CALL	CNS	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0104	3E 0D	0167	MVI	A,CR	
0106	12	0168	STAX	D	
0107	24 D3 48	0169	LXI	H,IBUF	
010A	CD 1E 0D	0170	CALL	PRNTR	
010D	C3 62 00	0171	JMP	CMND4	
0110		0172	*		
0110		0173	*	LINE EDITOR	
0110		0174	*		
0110	2A 50 19	0175	LINE	LHLD	BOFA
0113	7E	0176	FIN	MOV	A,M
0114	3D	0177		DCR	A
0115	CA 2F 01	0178		JZ	APP
0118	EB	0179		XCHG	
0119	13	0180		INX	D
011A	2A D1 18	0181		LHLD	IBLN
011D	EB	0182		XCHG	
011E	CD 35 0D	0183		CALL	DCMP
0121	2B	0184		DCX	H
0122	DA 44 01	0185		JC	INSR
0125	CA 44 01	0186		JZ	INSR
0128	7E	0187		MOV	A,M
0129	CD C1 01	0188		CALL	ADR
012C	C3 13 01	0189		JMP	FIN
012F		0190	*	APPEND LINE AT END CASE	
012F	3A D0 18	0191	APP	LDA	IBCNT
0132	FE 04	0192		CPI	4
0134	C8	0193		RZ	
0135	CD C6 01	0194		CALL	FULL
0138	2A 4E 19	0195		LHLD	EOFA
013B	CD 98 01	0196		CALL	IMOV
013E	36 01	0197		MVI	M,EOF
0140	22 4E 19	0198		SHLD	EOFA
0143	C9	0199		RET	
0144		0200	*	INSERT LINE IN FILE CASE	
0144	46	0201	INSR	MOV	B,M
0145	22 3D 19	0202		SHLD	INSA
0148	3A D0 18	0203		LDA	IBCNT
014B	DA 5C 01	0204		JC	LT
014E	D6 04	0205		SUI	4
0150	CA 55 01	0206		JZ	LT4
0153	C6 04	0207		ADI	4
0155	90	0208	LT4	SUB	B
0156	CA 8F 01	0209		JZ	LIN4
0159	DA 7B 01	0210		JC	GT
015C		0211	*	EXPAND FILE FOR NEW OR LARGER LINE	
015C	47	0212	LT	MOV	B,A
015D	3A D0 18	0213		LDA	IBCNT
0160	FE 04	0214		CPI	4
0162	C8	0215		RZ	
0163	78	0216		MOV	A,B
0164	CD C6 01	0217		CALL	FULL
0167	2A 3D 19	0218		LHLD	INSA
016A	CD B5 01	0219		CALL	NMOV
016D	2A 4E 19	0220		LHLD	EOFA
0170	EB	0221		XCHG	
0174	22 4E 19	0222		SHLD	EOFA

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0174 03	0223	INX	B
0175 CD AA 01	0224	CALL	RMOV
0178 C3 8F 01	0225	JMP	LIN1
017B	0226	* CONTRACT FILE FOR SMALLER LINE	
017B 2F	0227	GT	CMA
017C 3C	0228	INR	A
017D CD C1 01	0229	CALL	ADR
0180 CD B5 01	0230	CALL	NMOV
0183 EB	0231	XCHG	
0184 2A 3D 19	0232	LHLD	INSA
0187 C4 9F 01	0233	CNZ	LMOV
018A 36 01	0234	MVI	M,EOF
018C 22 4E 19	0235	SHLD	EOFA
018F	0236	* INSERT CURRENT LINE INTO FILE	
018F 2A 3D 19	0237	LIN1	LHLD INSA
0192 3A D0 18	0238	LDA	IBCNT
0195 FE 04	0239	CPI	4
0197 C8	0240	RZ	
0198	0241	* INSERT CURRENT LINE AT ADDR HL	
0198 11 D0 18	0242	IMOV	LXI D,IBCNT
019B 1A	0243	LDAX	D
019C 4F	0244	MOV	C,A
019D 06 00	0245	MVI	B,0
019F	0246	* COPY BLOCK FROM BEGINNING	
019F	0247	* HL IS DESTIN ADDR, DE IS SOURCE ADDR, BC IS COUNT	
019F 1A	0248	LMOV	LDAX D
01A0 77	0249	MOV	M,A
01A1 13	0250	INX	D
01A2 23	0251	INX	H
01A3 0B	0252	DCX	B
01A4 78	0253	MOV	A,B
01A5 B1	0254	ORA	C
01A6 C2 9F 01	0255	JNZ	LMOV
01A9 C9	0256	RET	
01AA	0257	* COPY BLOCK STARTING AT END	
01AA	0258	* HL IS DESTIN ADDR, DE IS SOURCE ADDR, BC IS COUNT	
01AA 1A	0259	RMOV	LDAX D
01AB 77	0260	MOV	M,A
01AC 2B	0261	DCX	H
01AD 1B	0262	DCX	D
01AE 0B	0263	DCX	B
01AF 78	0264	MOV	A,B
01B0 B1	0265	ORA	C
01B1 C2 AA 01	0266	JNZ	RMOV
01B4 C9	0267	RET	
01B5	0268	* COMPUTE FILE MOVE COUNT	
01B5	0269	* BC GETS (EOFA) - (HL), RET Z SET MEANS ZERO COUNT	
01B5 3A 4E 19	0270	NMOV	LDA EOFA
01B8 95	0271	SUB	L
01B9 4F	0272	MOV	C,A
01BA 3A 4F 19	0273	LDA	EOFA+1
01BD 9C	0274	SBB	H
01BE 47	0275	MOV	B,A
01BF B1	0276	ORA	C
01C0 C9	0277	RET	
01C1	0278	* ADD A TO HL	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

01C1 85	0279	ADR	ADD	L	
01C2 6F	0280		MOV	L,A	
01C3 D0	0281		RNC		
01C4 24	0282		INR	H	
01C5 C9	0283		RET		
01C6	0284	* CHECK FOR FILE OVERFLOW, LEAVES NEW EOFA IN DE			
01C6	0285	* A HAS INCREASE IN SIZE			
01C6 2A 4E 19	0286	FULL	LHLD	EOFA	
01C9 CD C1 01	0287		CALL	ADR	
01CC EB	0288		XCHG		
01CD 21 52 19	0289		LXI	H, MEMTOP	
01D0 CD 35 0D	0290		CALL	DCMP	
01D3 D2 F4 0C	0291		JNC	E8	
01D6 C9	0292		RET		
01D7	0293	*			
01D7	0294	*	COMMANDS		
01D7	0295	*			
01D7 2A 50 19	0296	CSCR	LHLD	BOFA	
01DA 36 01	0297		MVI	M, EOF	
01DC 22 4E 19	0298		SHLD	EOFA	
01DF	0299	* "CLEAR"			
01DF 2A 4E 19	0300	CCLEAR	LHLD	EOFA	CLEAR FROM EOFA TO MEMTOP
01E2 23	0301		INX	H	
01E3 22 1E 19	0302		SHLD	MATA	
01E6 EB	0303		XCHG		
01E7 21 52 19	0304		LXI	H, MEMTOP	END OF ASSIGNED MEMORY
01EA AF	0305	CCLR1	XRA	A	
01EB 12	0306		STAX	D	
01EC CD 35 0D	0307		CALL	DCMP	
01EF 13	0308		INX	D	
01F0 C2 EA 01	0309		JNZ	CCLR1	
01F3 2A 52 19	0310		LHLD	MEMTOP	
01F6 22 B3 18	0311		SHLD	STA	
01F9 21 FA 17	0312		LXI	H, CSTKL+CSTKSZ-1	
01FC 36 00	0313		MVI	M, ETYPE	
01FE 22 B5 18	0314		SHLD	CSTKA	
0201 21 B5 18	0315		LXI	H, ASTKL+ASTKSZ+FPSIZ-1	
0204 22 1C 19	0316		SHLD	ASTKA	
0207 C9	0317		RET		
0208	0318	* "NULL"			
0208 CD E0 0D	0319	CNULL	CALL	INTGER	
020B DA B0 00	0320		JC	E3	NO ARGUMENT SUPPLIED
020E 7D	0321		MOV	A,L	
020F 32 92 17	0322		STA	NULLCT	
0212 C3 62 00	0323		JMP	CMND1	
0215	0324	* "LIST"			
0215 CD 02 0D	0325	CLIST	CALL	GC	
0218 FE 0D	0326		CPI	CR	
021A 11 00 00	0327		LXI	D, 0	
021D CA 23 02	0328		JZ	CLO	JUMP IF NO ARG SUPPLIED
0220 CD E0 0D	0329		CALL	INTGER	ERROR DEFAULT IS LIST
0223 2A 50 19	0330	CLO	LHLD	BOFA	
0226 7E	0331	CL1	MOV	A,M	
0227 3D	0332		DCR	A	
0228 C8	0333		RZ		
0229 23	0334		INX	H	

022A CD 35 0D	0335	CALL	DCMP	
022D 2B	0336	DCX	H	POINT TO COUNT CHAR AGAIN
022E DA 3B 02	0337	JC	CL2	
0234 CA 3B 02	0338	JZ	CL2	
0234	0339	* INCREMENT TO NEXT LINE		
0234 7E	0340	MOV	A,M	
0235 CD C4 01	0341	CALL	ADR	
0238 C3 26 02	0342	JMP	CL4	
023B D5	0343	CL2	PUSH	D
023C 44 D3 18	0344	LXI	D,IBUF	AREA TO UNPREPROCESS TO
023F CD 09 07	0345	CALL	UPPL	
0242 23	0346	INX	H	
0243 E5	0347	PUSH	H	
0244 21 D3 18	0348	LXI	H,IBUF	
0247 CD 1E 0D	0349	CALL	PRNTR	
024A CD 26 10	0350	CALL	PCHECK	
024D CD 1C 10	0351	CALL	CRLF	
0250 E4	0352	POP	H	
0251 D1	0353	POP	D	
0252 C3 26 02	0354	JMP	CL4	
0255	0355	* "RUN"		
0255 CD DF 01	0356	CRUN	CALL	CCLEAR
0258 2A 50 19	0357	LHLD	BOFA	
025B 7E	0358	MOV	A,M	
025C 3D	0359	DCR	A	CHECK FOR NULL PROGRAM
025D CA 62 00	0360	JZ	END	
0260 23	0361	INX	H	
0264 23	0362	INX	H	
0262 23	0363	INX	H	
0263 22 95 17	0364	SHLD	TXA	
0266 22 B4 18	0365	SHLD	RTXA	POINTER FOR 'READ' STATEMENT
0269 AF	0366	XRA	A	
026A 32 94 17	0367	STA	DIRF	CLEAR DIRECT FLAG AND FALL THROUGH TO DRIVER
026D CD 1C 10	0368	CALL	CRLF	
0270	0369	*		
0270	0370	* INTERPRETER DRIVER		
0270	0371	*		
0270 CD 26 10	0372	ILOOP	CALL	PCHECK
0273 CD 7F 02	0373	CALL	ISTAT	INTERPRET CURRENT STATEMENT
0276 CD 2E 0C	0374	CALL	JOE	TEST FOR JUNK ON END
0279 D2 70 02	0375	JNC	ILOOP	CONTINUE IF NOT AT END OF PROGRAM
027C C3 62 00	0376	JMP	END	EXECUTE END STATEMENT
027F	0377	* INTERPRET STATEMENT LOCATED BY TXA		
027F CD 02 0D	0378	ISTAT	CALL	GC
0282 B7	0379	ORA	A	GET FIRST NON BLANK
0283 F2 9C 02	0380	JP	LET	MUST BE LET IF NOT RW
0286 FE 91	0381	CPI	IRWLM	IS IT AN INITIAL RW
0288 D2 AA 00	0382	JNC	E4	
028B 44 E0 07	0383	LXI	D,STATD	STATEMENT DISPATCH TABLE BASE
028E CD 0A 0D	0384	ISTA1	CALL	GCI
0291 E6 1F	0385	ANI	37Q	ADVANCE TEXT POINTER
0293 07	0386	RLC	.	
0294 6F	0387	MOV	L,A	MULTIPLY BY TWO PREPARING FOR TABLE LOOKUP
0295 26 00	0388	MVI	H,0	
0297 49	0389	DAD	D	
0298 CD 40 0D	0390	CALL	LHLI	

029B E9	0391	PCHL		BRANCH TO STATEMENT OR COMMAND
029C	0392 *			
029C	0393 *		STATEMENTS	
029C	0394 *			
029C	0395 * "LET"			
029C CD EA OB	0396 LET	CALL	VAR	CHECK FOR VARIABLE
029F DA AA 00	0397	JC	E4	
02A2 E5	0398	PUSH	H	SAVE VALUE ADDRESS
02A3 06 F5	0399	MVI	B,EQRW	
02A5 CD FA OC	0400	CALL	EATC	
02A8 CD D7 05	0401	CALL	EXPRB	
02AB D1	0402	POP	D	DESTINATION ADDRESS
02AC CD BE OC	0403	CALL	POPA4	COPY EXPR VALUE TO VARIABLE
02AF C9	0404	RET		
02B0	0405 * "FOR"			
02B0 CD 63 OD	0406 SFOR	CALL	DIRT	
02B3 CD EA OB	0407	CALL	VAR	CONTROL VARIABLE
02B6 DA AA 00	0408	JC	E4	
02B9 E5	0409	PUSH	H	CONTROL VARIABLE VALUE ADDRESS
02BA 06 F5	0410	MVI	B,EQRW	
02BC CD FA OC	0411	CALL	EATC	
02BF CD D7 05	0412	CALL	EXPRB	INITIAL VALUE
02C2 D1	0413	POP	D	VARIABLE VALUE ADDRESS
02C3 D5	0414	PUSH	D	SAVE
02C4 CD BE OC	0415	CALL	POPA4	SET INITIAL VALUE
02C7 06 9E	0416	MVI	B,TORW	RW FOR 'TO'
02C9 CD FA OC	0417	CALL	EATC	
02CC CD D7 05	0418	CALL	EXPRB	LIMIT VALUE COMPUTATION
02CF CD 02 OD	0419	CALL	GC	CHECK NEXT CHARACTER FOR POSSIBLE STEP EXPRESION
02D2 FE 9F	0420	CPI	STEPRW	
02D4 CA E0 02	0421	JZ	FOR1	
02D7	0422 * USE STEP OF 4			
02D7 11 87 07	0423	LXI	D,FPONE	
02DA CD A8 OC	0424	CALL	PSHA4	
02DD C3 E6 02	0425	JMP	FOR2	
02E0	0426 * COMPUTE STEP VALUE			
02E0 CD 0A OD	0427 FOR1	CALL	GCI	EAT THE STEP RW
02E3 CD D7 05	0428	CALL	EXPRB	THE STEP VALUE
02E6	0429 * HERE THE STEP AND LIMIT ARE ON ARG STACK			
02E6 11 FE FF	0430 FOR2	LXI	D,-2	PREPARE TO ALLOCATE 2 BYTES ON CONTROL STACK
02E9 CD CD OC	0431	CALL	PSHCS	RETURNS ADDRESS OF THOSE 2 BYTES IN HL
02EC EB	0432	XCHG		
02ED CD 2E OC	0433	CALL	JOE	TEST FOR JUNK ON END
02F0 DA B6 00	0434	JC	E4	NO "FOR" STATEMENT AT END OF PROGRAM
02F3 EB	0435	XCHG	.	DE HAS LOOP TEXT ADDR, HL HAS CONTROL STACK ADDR
02F4 72 "	0436	MOV	M,D	HIGH ORDER TEXT ADDRESS BYTE
02F5 2B	0437	DCX	H	
02F6 73	0438	MOV	M,E	LOW ORDER TEXT ADDRESS BYTE
02F7 11 FB FF	0439	LXI	D,-FPSIZ	ALLOCATE SPACE FOR LIMIT ON CONTROL STACK
02FA CD CD OC	0440	CALL	PSHCS	
02FD E5	0441	PUSH	H	ADDR ON CONTROL STACK FOR LIMIT
02FE 11 FB FF	0442	LXI	D,-FPSIZ	ALLOCATE SPACE FOR STEP ON CONTROL STACK
0301 CD CD OC	0443	CALL	PSHCS	
0304 CD BD OC	0444	CALL	POPAS	COPY STEP VALUE TO CONTROL STACK
0307 D1	0445	POP	D	CONTROL STACK ADDR FOR LIMIT VALUE
0308 CD BE OC	0446	CALL	POPA4	LIMIT VALUE TO CONTROL STACK

030B 14 FD FF	0447	LXI	D,-3	ALLOCATE SPACE FOR TEXT ADDRESS AND CS ENTRY
030E CD CD 0C	0448	CALL	PSHCS	
0311 D1	0449	POP	D	CONTROL VARIABLE ADDRESS
0312 72	0450	MOV	M,D	HIGH ORDER BYTE OF CONTROL VARIABLE ADDRESS
0313 2B	0451	DCX	H	
0314 73	0452	MOV	M,E	LOW ORDER BYTE OF CONTROL VARIABLE ADDRESS
0315 2B	0453	DCX	H	
0316 36 04	0454	MVI	M,FTYPE	SET CONTROL STACK ENTRY TYPE FOR 'FOR'
0318 C3 7C 03	0455	JMP	NEXT5	GO FINISH OFF CAREFULLY
031B	0456	* "NEXT"		
031B CD 63 0D	0457	CALL	DIRT	
031E 2A B5 18	0458	LHLD	CSTKA	CONTROL STACK ADDRESS
0321 7E	0459	MOV	A,M	STACK ENTRY TYPE BYTE
0322 3D	0460	DCR	A	MUST BE FOR TYPE ELSE ERROR
0323 C2 B6 00	0461	JNZ	E4	IMPROPER NESTING ERROR
0326 23	0462	INX	H	CONTROL STACK POINTER TO CONTROL VARIABLE ADDRESS
0327 E5	0463	PUSH	H	
0328 CD EA 0B	0464	CALL	VAR	CHECK VARIABLE, IN CASE USER WANTS
032B DA 37 03	0465	JC	NEXT4	SKIP CHECK IF VAR NOT THERE
032E EB	0466	XCHG		
032F E1	0467	POP	H	CONTROL VARIABLE ADDRESS
0330 E5	0468	PUSH	H	SAVE IT AGAIN
0331 CD 35 0D	0469	CALL	DCMP	
0334 C2 B6 00	0470	JNZ	E4	IMPROPER NESTING IF NOT THE SAME
0337 E1	0471	POP	H	CONTROL VARIABLE ADDRESS
0338 E5	0472	PUSH	H	
0339 E5	0473	PUSH	H	
033A 11 06 00	0474	LXI	D,FPSIZ+2-1	COMPUTE ADDRESS TO STEP VALUE
033D 19	0475	DAD	D	
033E E3	0476	XTHL	.	NOW ADDRESS TO VAR IN HL
033F CD 40 0D	0477	CALL	LHLI	VARIABLE ADDRESS
0342 44	0478	MOV	B,H	COPY VAR ADDRESS TO BC
0343 4D	0479	MOV	C,L	
0344 D1	0480	POP	D	STEP VALUE ADDRESS
0345 D5	0481	PUSH	D	
0346 CD C8 13	0482	CALL	FADD	DO INCREMENT
0349 E1	0483	POP	H	STEP VALUE
034A 2B	0484	DCX	H	POINT TO SIGN OF STEP VALUE
034B 7E	0485	MOV	A,M	SIGN 0=POS, 1=NEG
034C 14 06 00	0486	LXI	D,FPSIZ+4	
034F 19	0487	DAD	D	PUTS LIMIT ADDRESS IN HL
0350 EB	0488	XCHG		
0351 E1	0489	POP	H	VARIABLE ADDRESS
0352 CD 40 0D	0490	CALL	LHLI	GET ADDRESS
0355 D5	0491	PUSH	D	SAVE CONTROL STACK POINTER TO GET TEXT ADDRESS
0356 B7	0492	ORA	A	SET CONDITIONS BASED ON SIGN OF STEP VALUE
0357 CA 5B 03	0493	JZ	NEXT2	REVERSE TEST ON NEGATIVE STEP VALUE
035A EB	0494	XCHG		
035B 44	0495	MOV	B,H	SET UP ARGS FOR COMPARE
035C 4D	0496	MOV	C,L	
035D CD 64 09	0497	CALL	RELOP	TEST <=
0360 J1	0498	POP	D	TEXT ADDRESS
0361 FA 6F 03	0499	JM	NEXT3	STILL SMALLER?
0364 CA 6F 03	0500	JZ	NEXT3	JUMP IF WANT TO CONTINUE LOOP
0367	0501	* TERMINATE LOOP		
0367 21 03 00	0502	LXI	H,3	REMOVE CSTACK ENTRY

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

036A 19	0503	DAD	D	
036B 22 B5 18	0504	SHLD	CSTKA	
036E C9	0505	RET		
036F 13	0506 NEXT3	INX	D	TEXT ADDRESS
0370 EB	0507	XCHG		
0371 CD 40 0D	0508	CALL	LHLI	GET TEXT ADDRESS IN HL
0374	0509	* ITERATE, SKIPPING NORMAL JUNK ON END TEST AT ILOOP		
0374 EB	0510 NEXT4	XCHG	.	SAVE NEW TEXT ADDRESS IN DE
0375 CD 2E 0C	0511	CALL	JOE	
0378 EB	0512	XCHG		
0379 22 95 1'	0513 NEXT6	SHLD	TXA	
037C 21 70 02	0514 NEXT5	LXI	H,ILOOP	
037F E3	0515	XTHL		
0380 C9	0516	RET	.	TO DISPATCHER SKIPPING JOE CALL THERE
0381	0517	* "IF"		
0381 06 01	0518 SIF	MVI	B,1	SPECIFY PRINCIPAL OPERATOR IS RELATIONAL
0383 CD D9 05	0519	CALL	EXPB1	
0386 2A 1C 19	0520	LHLD	ASTKA	ADDRESS OF BOLLEAN VALUE ON ARG STACK
0389 34	0521	INR	M	SETS ZERO CONDITION IF RELATIONAL WAS TRUE
038A F5	0522	PUSH	PSW	SAVE CONDITIONS TO TEST LATER
038B CD BD 0C	0523	CALL	POPAS	REMOVE VALUE FROM ARG STACK COPY TO SELF
038E F1	0524	POP	PSW	
038F C2 08 04	0525	JNZ	REM	IF TEST FALSE TREAT REST OF STATEMENT AS REM
0392	0526	* TEST SUCCEEDED		
0392 06 9D	0527	MVI	B,THENRW	
0394 CD FA 0C	0528	CALL	EATC	
0397 CD E0 0D	0529	CALL	INTGER	CHECK IF LINE NUMBER IS DESIRED ACTION
039A DA 7F 02	0530	JC	ISTAT	
039D C3 AA 03	0531	JMP	GOTO1	
03A0	0532	* "GOTO"		
03A0 AF	0533 SGOTO	XRA	A	
03A1 32 94 17	0534	STA	DIRF	CLEAR DIRECT STATEMENT FLAG
03A4 CD E0 0D	0535	CALL	INTGER	RETURNS INTEGER IN HL IF LINE NUMBER PRESENT
03A7 DA AA 00	0536	JC	E1	SYNTAX ERROR NO LINE NUMBER
03AA EB	0537 GOTO1	XCHG	.	LN IN DE
03AB CD 6E 0D	0538	CALL	FINDLN	RETURNS TEXT ADDRESS POINTS TO COUNT VALUE
03AE 23	0539 GOTO2	INX	H	
03AF 23	0540	INX	H	
03B0 23	0541	INX	H	ADVANCE TEXT POINTER PAST LINE NUMBER AND COUNT
03B1 C3 74 03	0542	JMP	NEXT4	
03B4	0543	* "GOSUB"		
03B4 CD 63 0D	0544 GOSUB	CALL	DIRT	
03B7 11 FD FF	0545	LXI	D,-3	CREATE CONTROL STACK ENTRY
03BA CD CD 0C	0546	CALL	PSHCS	
03BD E5 ..	0547	PUSH	H	SAVE STACK ADDRESS
03BE CD E0 0D	0548	CALL	INTGER	
03C1 DA AA 00	0549	JC	E1	
03C4 EB	0550	XCHG	.	LINE NUMBER TO DE
03C5 CD 2E 0C	0551	CALL	JOE	
03C8 44	0552	MOV	B,H	
03C9 4D	0553	MOV	C,L	
03CA E1	0554	POP	H	STACK ADDRESS
03CB 70	0555	MOV	M,B	STACK RETURN ADDRESS RETURNED BY JOE
03CC 2B	0556	DCX	H	
03CD 71	0557	MOV	M,C	
03CE 2B	0558	DCX	H	

03CF 36 02	0559	MVI	M,GTYPE	MAKE CONTROL STACK ENTRY TYPE 'GOSUB'
03D1 CD 6E 0D	0560	CALL	FINDLN	
03D4 23	0564	INX	H	
03D5 23	0562	INX	H	
03D6 23	0563	INX	H	
03D7 C3 79 03	0564	JMP	NEXT6	
03DA	0565	* "RETURN"		
03DA CD 63 0D	0566	RETRN	CALL	DIRT
03DD 32 94 17	0567	STA	DIRF	CLEAR DIRF IF ACC IS CLEAR
03E0 2A B5 18	0568	LHLD	CSTKA	
03E3 7E	0569	RET1	MOV	A,M
03E4 B7	0570	ORA	A	CHECK FOR STACK EMPTY
03E5 CA B6 00	0571	JZ	E4	
03E8 FE 02	0572	CPI	GTYPE	CHECK FOR GOSUB TYPE
03EA CA F4 03	0573	JZ	RET2	
03ED	0574	* REMOVE FOR TYPE ENTRY FROM STACK		
03ED 11 0F 00	0575	LXI	D,FORSZ	
03F0 19	0576	DAD	D	
03F1 C3 E3 03	0577	JMP	RET4	
03F4	0578	* FOUND A GTYPE STACK ENTRY		
03F4 23	0579	RET2	INX	H
03F5 5E	0580	MOV	E,M	LOW ORDER TEXT ADDRESS
03F6 23	0581	INX	H	
03F7 56	0582	MOV	D,M	HIGH ORDER TEXT ADDRESS
03F8 23	0583	INX	H	ADDRESS OF PREVIOUS CONTROL STACK ENTRY
03F9 22 B5 18	0584	SHLD	CSTKA	
03FC EB	0585	XCHG	.	PUT TEXT ADDRESS IN HL
03FD 7E	0586	MOV	A,M	ADDRESS POINTS TO EOF IF GOSUB WAS LAST LINE
03FE 3D	0587	DCR	A	END OF FILE?
03FF C2 74 03	0588	JNZ	NEXT4	
0402 C3 62 00	0589	JMP	END	
0405	0590	* "DATA AND REM"		
0405 CD 63 0D	0594	DATA	CALL	DIRT
0408 CD 0A 0D	0592	REM	CALL	GCI
040B FE 0D	0593	CPI	CR	
040D C2 08 04	0594	JNZ	REM	
0410 2B	0595	DCX	H	BACKUP POINTER SO NORMAL JOE WILL WORK
0411 22 95 17	0596	SHLD	TXA	
0414 C9	0597	RET		
0415	0598	* "DIMENSION"		
0415 CD 49 0C	0599	DIM	CALL	NAME
0418 DA AA 00	0600	JC	E4	LOOK FOR VARIABLE NAME
041B 79	0601	MOV	A,C	PREPARE TURN ON 200Q BIT TO SIGNIFY MATRIX
041C F6 80	0602	ORI	200Q	
041E 4F	0603	MOV	C,A	
041F CD 58 0C	0604	CALL	STLK	
0422 D2 C2 00	0605	JNC	E6	ERROR IF NAME ALREADY EXISTS
0425 E5	0606	PUSH	H	SYMBOL TABLE ADDRESS
0426 06 E0	0607	MVI	B,LPARRW	
0428 CD FA 0C	0608	CALL	EATC	
042B CD D7 05	0609	CALL	EXPRB	
042E 06 29	0610	MVI	B,')'	
0430 CD FA 0C	0611	CALL	EATC	
0433 CD 8A 0D	0612	CALL	PFIX	RETURN INTEGER IN DE
0436 24 AC 07	0613	LXI	H,MATUB	MAXIMUM SIZE FOR MATRIX
0439 CD 35 0D	0614	CALL	DCMP	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

043C D2 C2 00	0645	JNC	E6	
043F E4	0646	POP	H	SYMBOL TABLE ADDRESS
0440 CD C6 0B	0647	CALL	DIMS	
0443 CD 02 0D	0648	CALL	GC	SEE IF MORE TO DO
0446 FE 2C	0649	CPI	','	
0448 C0	0620	RNZ		
0449 CD 0A 0D	0624	CALL	GCI	EAT THE COMMA
044C C3 45 04	0622	JMP	DIM	
044F	0623	* "STOP"		
044F CD 63 0D	0624	STOP	CALL	DIRT
0452 CD 19 10	0625	STOP4	CALL	CRLF2
0455 32 3F 49	0626		STA	BRKCHR
0458 24 65 07	0627		LXI	H,STOPS
045B C3 D4 00	0628		JMP	ERM1
045E	0629	* "END"		
045E	0630	END	EQU	CMND4
045E	0634	* "READ"		
045E CD 63 0D	0632	READ	CALL	DIRT
0464 2A 95 17	0633		LHLD	TXA
0464 E5	0634		PUSH	H
0465 2A B4 18	0635		LHLD	RTXA
0468 22 95 17	0636	READO	SHLD	TXA
046B CD 0A 0D	0637		CALL	GCI
046E FE 2C	0638		CPI	','
0470 CA 8B 04	0639		JZ	READ2
0473 FE 87	0640		CPI	DATARW
0475 CA 8B 04	0641		JZ	READ2
0478 3D	0642		DCR	A
0479 CA B8 04	0643		JZ	READ4
047C	0644	* SKIP TO NEXT LINE		
047C CD 08 04	0645		CALL	REM
047F 23	0646		INX	H
0480 7E	0647		MOV	A,M
0481 3D	0648		DCR	A
0482 CA B8 04	0649		JZ	READ4
0485 23	0650		INX	H
0486 23	0651		INX	H
0487 23	0652		INX	H
0488 C3 68 04	0653		JMP	READO
048B	0654	* PROCESS VALUE		
048B CD D7 05	0655	READ2	CALL	EXPRB
048E CD 02 0D	0656		CALL	GC
0491 FE 2C	0657		CPI	','
0493 CA 99 04	0658		JZ	READ3
0496	0659	* JUNK ON END TEST		
0496 CD 2E 0C	0660		CALL	JOE
0499 2A 95 17	0664	READ3	LHLD	TXA
049C 22 B4 18	0662		SHLD	RTXA
049F E4	0663		POP	H
04A0 22 95 17	0664		SHLD	TXA
04A3 CD EA 0B	0665		CALL	VAR
04A6 DA AA 00	0666		JC	E4
04A9 CD BD 0C	0667		CALL	POPAS
04AC CD 02 0D	0668		CALL	GC
04AF FE 2C	0669		CPI	','
04B1 C0	0670		RNZ	
				CHECK FOR ANOTHER VARIABLE.
				PUT READ VALUE INTO VARIABLE
				SAVE NEW "READ" TEXT ADDRESS
				REAL TXA
				SAVE TXA TEMPORARILY
				THE 'READ' TXA
				PROCESS INPUT VALUE
				LEAVES ADDRESS OF LAST CR IN HL
				HL NOW POINTS TO FIRST BYTE OF NEXT LINE

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT #976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

04B2 CD 0A 0D	0674	CALL	GCI	EAT THE COMMA
04B5 C3 5E 04	0672	JMP	READ	
04B8 E4	0673	READ4	POP	PROGRAM TXA
04B9 22 95 17	0674		SHLD	TXA
04BC 01 44 52	0675		LXI	B,'RD'
04BF C3 C5 00	0676		JMP	ERROR
04C2	0677	* "RESTORE"		
04C2 2A 50 19	0678	RESTOR	LHLD	BOFA
04C5 23	0679		INX	H
04C6 23	0680		INX	H
04C7 23	0681		INX	H
04C8 22 B1 18	0682		SHLD	RTXA
04CB C9	0683		RET	
04CC	0684	* "PRINT"		
04CC CD 02 0D	0685	PRINT	CALL	GC
04CF FE 0D	0686		CPI	CR
04D1 CA 1C 40	0687		JZ	CRLF
04D4 FE 22	0688	PRIN2	CPI	'"
04D6 CA 16 05	0689		JZ	PSTR
04D9 FE 9C	0690		CPI	TABRW
04DB CA 58 05	0691		JZ	PTAB
04DE FE 25	0692		CPI	'%
04E0 CA 23 05	0693		JZ	PFORM
04E3 FE 0D	0694		CPI	CR
04E5 C8	0695		RZ	
04E6 FE 3B	0696		CPI	';
04E8 C8	0697		RZ	
04E9 CD D7 05	0698		CALL	EXPRB
04EC 14 BB 18	0699		LXI	D,FPSINK
04EF CD BE 0C	0700		CALL	POPA4
04F2 3A 90 17	0701		LDA	PHEAD
04F5 FE 38	0702		CPI	56
04F7 D4 1C 10	0703		CNC	CRLF
04FA 24 BB 18	0704		LXI	H,FPSINK
04FD CD 73 40	0705		CALL	FPOUT
0500 06 20	0706		MVI	B,' '
0502 CD 95 0E	0707		CALL	CHOUT
0505 CD 02 0D	0708	PR4	CALL	GC
0508 FE 2C	0709		CPI	' ,'
050A C2 1C 10	0710		JNZ	CRLF
050D CD 0A 0D	0711	PRO	CALL	GCI
0510 CD 02 0D	0712		CALL	GC
0513 C3 D4 04	0713		JMP	PRIN2
0516 CD 0A 0D	0714	PSTR	CALL	GCI
0519 CD 23 0D	0715		CALL	PRNT
051C 23	0716		INX	H
051D 22 95 17	0717		SHLD	TXA
0520 C3 05 05	0718		JMP	PR4
0523 3E 0C	0719	PFORM	MVI	A,2*FPNIB
0525 32 3A 19	0720		STA	INFES
0528 CD 0A 0D	0721		CALL	GCI
052B CD 0A 0D	0722	PFRM1	CALL	GCI
052E 24 3A 19	0723		LXI	H,INFES
0531 FE 25	0724		CPI	'%
0533 CA 05 05	0725		JZ	PR4
0536 06 80	0726		MVI	B,200Q

0538 FE 5A	0727	CPI	'Z'	TRAILING ZEROS?
053A CA 52 05	0728	JZ	PF1	
053D 06 01	0729	MVI	B,1	
053F FE 45	0730	CPI	'E'	SCIENTIFIC NOTATION?
0541 CA 52 05	0731	JZ	PF0	
0544 CD FA 12	0732	CALL	NMCHK	
0547 D2 AA 00	0733	JNC	E1	
054A D6 30	0734	SUI	'0'	NUMBER OF DECIMAL PLACES
054C 07	0735	RLC		
054D 47	0736	MOV	B,A	
054E 7E	0737	MOV	A,M	
054F E6 C1	0738	ANI	301Q	
0551 77	0739	MOV	M,A	
0552 7E	0740 PF1	MOV	A,M	
0553 B0	0741	ORA	B	
0554 77	0742	MOV	M,A	
0555 C3 2B 05	0743	JMP	PFRM1	
0558 CD 0A 0D	0744 PTAB	CALL	GCI	GOBBLE TAB RW
055B 06 E0	0745	MVI	B,LPARRW	
055D CD FA 0C	0746	CALL	EATC	
0560 CD D7 05	0747	CALL	EXPRB	
0563 06 29	0748	MVI	B,')'	
0565 CD FA 0C	0749	CALL	EATC	
0568 CD 8A 0D	0750	CALL	PFIX	
056B 3A 90 17	0751 PTAB1	LDA	PHEAD	
056E BB	0752	CMP	E	
056F D2 05 05	0753	JNC	PR1	
0572 06 20	0754	MVI	B,' '	
0574 CD 95 0E	0755	CALL	CHOUT	
0577 C3 6B 05	0756	JMP	PTAB1	
057A	0757 * "INPUT"			
057A CD 02 0D	0758 INPUT	CALL	GC	
057D FE 2C	0759	CPI	',,'	
057F CA CB 05	0760	JZ	NCRLF	
0582 CD 1C 10	0761	CALL	CRLF	
0585 06 3F	0762 INPO	MVI	B,'?'	
0587 CD 95 0E	0763	CALL	CHOUT	
058A CD 55 0E	0764 LINP	CALL	INLINE	
058D 11 D3 18	0765	LXI	D,IBUF	
0590 D5	0766 IN1	PUSH	D	SAVE FOR FPIN
0591 CD EA 0B	0767	CALL	VAR	
0594 DA AA 00	0768	JC	E1	
0597 D1	0769	POP	D	
0598 06 00	0770	MVI	B,0	
059A 1A	0771	LDAX	D	
059B FE 2B	0772	CPI	'+'	LOOK FOR LEADING PLUS OR MINUS ON INPUT
059D CA A7 05	0773	JZ	IN2	
05A0 FE 2D	0774	CPI	'-'	
05A2 C2 A8 05	0775	JNZ	IN3	
05A5 06 01	0776	MVI	B,1	
05A7 13	0777 IN2	INX	D	
05A8 C5	0778 IN3	PUSH	B	
05A9 E5	0779	PUSH	H	
05AA CD 57 12	0780	CALL	FPIN	INPUT FP NUMBER
05AD DA D1 05	0781	JC	INERR	
05B0 E1	0782	POP	H	

05B1 2B	0783	DCX	H	
05B2 F1	0784	POP	PSW	
05B3 77	0785	MOV	M,A	
05B4 CD 02 0D	0786	CALL	GC	
05B7 FE 2C	0787	CPI	' ,'	
05B9 C0	0788	RNZ	.	DONE IF NO MORE
05BA CD 0A 0D	0789	CALL	GCI	EAT THE COMMA
05BD 78	0790	MOV	A,B	GET THE TERMINATOR TO A
05BE FE 2C	0791	CPI	' ,'	
05C0 CA 90 05	0792	JZ	IN1	GET THE NEXT INPUT VALUE FROM STRING
05C3	0793	* GET NEW LINE FROM USER		
05C3 06 3F	0794	MVI	B,'?'	
05C5 CD 95 0E	0795	CALL	CHOUT	
05C8 C3 85 05	0796	JMP	INPO	
05CB CD 0A 0D	0797	NCRLF	CALL	GCI
05CE C3 8A 05	0798	JMP	LINP	NOW GET LINE
05D1 01 4E 49	0799	INERR	LXI	B,'IN'
05D4 C3 C5 00	0800	JMP	ERROR	
05D7	0801	*		
05D7	0802	* EVALUATE AN EXPRESSION FROM TEXT		
05D7	0803	* HL TAKE OP TABLE ADDR OF PREVIOUS OPERATOR (NOT CHANGED)		
05D7	0804	* RESULT VALUE LEFT ON TOP OF ARG STACK, ARGF LEFT TRUE		
05D7	0805	*		
05D7 06 00	0806	EXPRB	MVI	B,0
05D9 24 EE 08	0807	EXPR1	LXI	H,OPBOL
05DC AF	0808		XRA	A
05DD 32 91 17	0809		STA	RELTYP
05E0	0810	* ZERO IN B MEANS PRINCIPAL OPERATOR MAY NOT BE RELATIONAL		
05E0 C5	0811	EXPR	PUSH	B
05E1 E5	0812		PUSH	H
05E2 AF	0813		XRA	A
05E3 32 93 17	0814		STA	ARGF
05E6 3A 93 17	0815	EXPR1	LDA	ARGF
05E9 B7	0816		ORA	A
05EA C2 07 06	0817		JNZ	EXPR2
05ED CD EA 0B	0818		CALL	VAR
05F0 D4 A7 0C	0819		CNC	PSHAS
05F3 D2 07 06	0820		JNC	EXPR2
05F6 CD 47 0D	0821		CALL	CONST
05F9 D2 07 06	0822		JNC	EXPR2
05FC CD 02 0D	0823		CALL	GC
05FF FE E0	0824		CPI	LPARRW
0601 24 D6 08	0825		LXI	H,OPLPAR
0604 CA 83 06	0826		JZ	XLPAR
0607	0827	* ISN'T OR SHOULDN'T BE AN ARGUMENT		
0607 CD 02 0D	0828	EXPR2	CALL	GC
060A FE E0	0829		CPI	340Q
060C D2 25 06	0830		JNC	XOP
060F FE C0	0831		CPI	300Q
0611 D2 74 06	0832		JNC	XBILT
0614	0833	* ILLEGAL EXPRESSION CHARACTER		
0614 E1	0834		POP	H
0615 3A 93 17	0835		LDA	ARGF
0618 B7	0836		ORA	A
0619 CA AA 00	0837		JZ	E1
061C F1	0838	XDON1	POP	PSW

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

061D 21 91 17	0839	LXI	H,RELTYP	CHECK IF LEGAL PRINCIPAL OPERATOR
0620 BE	0840	CMP	M	
0621 C8	0841	RZ		
0622 C3 AA 00	0842	JMP	E1	
0625 E6 1F	0843 XOP	ANI	37Q	CLEANS OFF RW BITS
0627 2A 93 17	0844	LHLD	ARGF	TEST FOR ARGF TRUE
062A 2D	0845	DCR	L	
062B CA 40 06	0846	JZ	XOP1	
062E	0847 *	ARGF WAS FALSE,	UNARY OPS ONLY POSSIBILITY	
062E FE 05	0848	CPI	'-'-OPBASE	
0630 CA 3E 06	0849	JZ	XOPM	
0633 FE 03	0850	CPI	'+'-OPBASE	
0635 C2 AA 00	0851	JNZ	E1	
0638 CD 0A 0D	0852	CALL	GCI	EAT THE '+'
063B C3 E6 05	0853	JMP	EXPR1	
063E 3E 09	0854 XOPM	MVI	A,UMINUS-OPBASE	
0640 CD 95 06	0855 XOP1	CALL	OPADR	
0643 D1	0856	POP	D	PREVIOUS OPTBA
0644 1A	0857	LDAX	D	
0645 BE	0858	CMP	M	
0646 D2 1C 06	0859	JNC	XDON1	NON-INCREASING PRECEDENCE
0649	0860 *	INCREASING PRECEDENCE CASE		
0649 D5	0861	PUSH	D	SAVE PREVIOUS OPTBA
064A E5	0862	PUSH	H	SAVE CURRENT OPTBA
064B CD 0A 0D	0863	CALL	GCI	TO GOBBLE OPERATOR
064E E1	0864	POP	H	
064F E5	0865	PUSH	H	
0650 06 00	0866	MVI	B,0	SPECIFY NON-RELATIONAL
0652 CD E0 05	0867	CALL	EXPR	
0655 E1	0868	POP	H	
0656	0869 *	HL HAS OPTBA ADDRESS		
0656	0870 *	SET UP ARGS AND PERFORM OPERATION ACTION		
0656 E5	0871 XOP2	PUSH	H	
0657 7E	0872	MOV	A,M	
0658 2A 1C 19	0873	LHLD	ASTKA	
065B 44	0874	MOV	B,H	
065C 4D	0875	MOV	C,L	
065D E6 01	0876	ANI	1	
065F C2 6B 06	0877	JNZ	XOP21	
0662	0878 *	DECREMENT STACK POINTER BY ONE VALUE BINARY CASE		
0662 11 05 00	0879	LXI	D,FPSIZ	
0665 19	0880	DAD	D	
0666 22 1C 19	0881	SHLD	ASTKA	
0669 54	0882	MOV	D,H	
066A 5D	0883	MOV	E,L	
066B 21 E6 05	0884 XOP21	LXI	H,EXPR1	
066E E3	0885	XTHL	.	CHANGE RETURN LINK
066F 23	0886	INX	H	SKIP OVER PRECEDENCE
0670 CD 40 0D	0887	CALL	LHLI	LOAD ACTION ADDRESS
0673 E9	0888	PCHL		
0674	0889 *			
0674	0890 *	ACTION ROUTINE CONVENTION		
0674	0891 *	DE LEFT ARG AND RESULT FOR BINARY		
0674	0892 *	BC RIGHT ARG FOR BINARY, ARG AND RESULT FOR UNARY		
0674	0893 *	BUILT IN FUNCTION PROCESSING		
0674	0894 *			

0674	CD	0A	0D	0895	XBILT	CALL	GCI	EAT	TOKEN
0677	E6	3F		0896		ANI	77Q	CLEAN	OFF RW BITS
0679	2A	93	17	0897		LHLD	ARGF	BUILT	IN FUNCTION MUST COME AFTER OPERATOR
067C	2D			0898		DCR	L		
067D	CA	AA	00	0899		JZ	E4		
0680	CD	95	06	0900		CALL	OPADR	OPTBA	TO HL
0683	E5			0901	XLPAR	PUSH	H		
0684	06	E0		0902		MVI	B,LPARRW		
0686	CD	FA	0C	0903		CALL	EATC		
0689	CD	D7	05	0904		CALL	EXPRB		
068C	06	29		0905		MVI	B,')'		
068E	CD	FA	0C	0906		CALL	EATC		
0691	E4			0907		POP	H	CODE	FOR BUILT-IN FUNTION
0692	C3	56	06	0908		JMP	XOP2		
0695				0909	*	COMPUTE	OPTABLE	ADDRESS	FOR OPERATOR IN ACC
0695	4F			0910	OPADR	MOV	C,A		
0696	06	00		0911		MVI	B,0		
0698	24	D6	08	0912		LXI	H,OPTAB		
069B	09			0913		DAD	B		
069C	09			0914		DAD	B		
069D	09			0915		DAD	B	OPTAB	ENTRY ADDR IS 3*OP+BASE
069E	C9			0916		RET			
069F				0917	*				
069F				0918	*	PREPROCESSOR,	UN-PREPROCESSOR		
069F				0919	*	PREPROCESS	LINE IN IBUF	BACK	INTO IBUF
069F				0920	*	SETS	CARRY IF	LINE	HAS NO LINE NUMBER
069F				0921	*	LEAVES	CORRECT	LENGTH	OF LINE AFTER PREPROCESSING IN IBCN
069F				0922	*	IF	THERE	IS	A LINE NUMBER, IT IS LOCATED AT IBLN=IBUF-2
069F				0923	*	TXA	IS	CLOBBBERED	
069F				0924	*				
069F	24	D3	18	0925	PP	LXI	H,IBUF	FIRST	CHARACTER OF INPUT LINE
06A2	22	95	17	0926		SHLD	TXA	SO	GCI WILL WORK
06A5	CD	E0	0D	0927		CALL	INTGER	SETS	CARRY IF NO LINE NUMBER
06A8	22	D4	18	0928		SHLD	IBLN	STORE	LINE NUMBER VALUE (EVEN IF NONE)
06AB	F5			0929		PUSH	PSW	SAVE	STATE OF CARRY BIT FOR RETURNING
06AC	2A	95	17	0930		LHLD	TXA	ADDRESS	OF NEXT CHARACTER IN IBUF
06AF	0E	04		0931		MVI	C,4	SET	UP INITIAL VALUE FOR COUNT
06B1	14	D3	18	0932		LXI	D,IBUF	INITIALIZE	WRITE POINTER
06B4				0933	*	COME	HERE	TO	CONTINUE PREPROCESSING LINE
06B4	D5			0934	PPL	PUSH	D		
06B5	14	02	08	0935		LXI	D,RWT	BASE	OF RWT
06B8	E5			0936	PPL4	PUSH	H	SAVE	TEXT ADDRESS
06B9	1A			0937		LDAX	D	RW	VALUE FOR THIS ENTRY IN RWT
06BA	47			0938		MOV	B,A	SAVE	IN B IN CASE OF MATCH
06BB	13			0939	PPL2	INX	D	ADVANCE	ENTRY POINTER TO NEXT BYTE
06BC	1A			0940		LDAX	D	GET	NEXT CHARACTER FROM ENTRY
06BD	BE			0941		CMP	M	COMPARE	WITH CHARACTER IN TEXT
06BE	C2	C5	06	0942		JNZ	PPL3		
06C1	23			0943		INX	H	ADVANCE	TEXT POINTER
06C2	C3	BB	06	0944		JMP	PPL2	CONTINUE	COMPARISON
06C5				0945	*	COME	HERE	WHEN	COMPARISON OF BYTE FAILED
06C5	B7			0946	PPL3	ORA	A		
06C6	FA	F7	06	0947		JM	PPL6	JUMP	IF FOUND MATCH
06C9				0948	*	SCAN	TO	BEGINNING	OF NEXT ENTRY
06C9	13			0949	PPL4	INX	D	ADVANCE	ENTRY POINTER
06CA	1A			0950		LDAX	D	NEXT	BYTE IS EITHER CHARACTER OR RW BYTE

06CB B7	0951	ORA	A	
06CC F2 C9 06	0952	JP	PPL4	KEEP SCANNING IF NOT RW BYTE
06CF	0953	* NOW SEE IF AT END OF TABLE, AND FAIL OR RETURN CONDITION		
06CF E1	0954	POP	H	RECOVER ORIGINAL TEXT POINTER
06D0 EE FF	0955	XRI	377Q	CHECK FOR END OF TABLE BYTE
06D2 C2 B8 06	0956	JNZ	PPL1	CONTINUE SCAN OF TABLE
06D5	0957	* DIDN'T FIND AN ENTRY AT THE GIVEN TEXT ADDR		
06D5 D1	0958	POP	D	
06D6 7E	0959	MOV	A,M	GET TEXT CHARACTER
06D7 FE 0D	0960	CPI	CR	CHECK FOR END OF LINE
06D9 CA 00 07	0961	JZ	PPL8	GO CLEAN UP AND RETURN
06DC 12	0962	STAX	D	
06DD 13	0963	INX	D	
06DE 0C	0964	INR	C	
06DF 23	0965	INX	H	ADVANCE TEXT POINTER
06E0 FE 22	0966	CPI	'"'	CHECK FOR QUOTED STRING POSSIBILITY
06E2 C2 B4 06	0967	JNZ	PPL	RESTART RWT SEARCH AT NET CHARACTER POSITION
06E5	0968	* HERE WE HAVE A QUOTED STRING, SO EAT TILL ENDQUOTE		
06E5 7E	0969	PPL5	MOV	A,M
06E6 FE 0D	0970	CPI	CR	NEXT CHARACTER
06E8 CA 00 07	0971	JZ	PPL8	NO STRING ENDQUOTE, LET INTERPRETER WORRY
06EB 12	0972	STAX	D	
06EC 13	0973	INX	D	
06ED 0C	0974	INR	C	
06EE 23	0975	INX	H	ADVANCE TEXT POINTER
06EF FE 22	0976	CPI	'"'	
06F1 CA B4 06	0977	JZ	PPL	BEGIN RWT SCAN FROM NEW CHARACTER POSITION
06F4 C3 E5 06	0978	JMP	PPL5	
06F7	0979	* FOUND MATCH SO PUT RW VALUE IN TEXT		
06F7 F1	0980	PPL6	POP	PSW
06F8 D1	0981	POP	D	REMOVE UNNEEDED TEST POINTER FROM STACK
06F9 78	0982	MOV	A,B	
06FA 12	0983	STAX	D	
06FB 13	0984	INX	D	
06FC 0C	0985	INR	C	
06FD C3 B4 06	0986	JMP	PPL	
0700	0987	* COME HERE WHEN DONE		
0700 3E 0D	0988	PPL8	MVI	A,CR
0702 12	0989	STAX	D	
0703 21 D0 18	0990	LXI	H,IBCNT	SET UP COUNT IN CASE LINE OF LINE NUMBER
0706 71	0991	MOV	M,C	
0707 F1	0992	POP	PSW	RESTORE CARRY CONDITION (LINE NUMBER FLAG)
0708 C9	0993	RET		
0709	0994	*		
0709	0995	* UN PREPROCESS LINE ADDRESSES IN HL TO DE BUFFER		
0709	0996	* RETURN SOURCE ADDRESS OF CR IN HL ON RETURN		
0709	0997	*		
0709 23	0998	UPPL	INX	H
070A E5	0999	PUSH	H	SAVE SOURCE TEXT POINTER
070B CD 40 0D	1000	CALL	LHLI	LOAD LINE NUMBER VALUE
070E CD FA 0D	1001	CALL	CNS	CONVERT LINE NUMBER
0711 3E 20	1002	MVI	A,''	
0713 12	1003	STAX	D	PJT BLANK AFTER LINE NUMBER
0714 13	1004	INX	D	INCREMENT DESTINATION POINTER
0715 E1	1005	POP	H	
0716 23	1006	INX	H	INCREMENT H PAST LINE NUMBER

0717 23	4007 UPPO	INX	H	
0718 7E	4008	MOV	A,M	NEXT TOKEN IN SOURCE
0719 B7	4009	ORA	A	
071A FA 25 07	4010	JM	UPP1	JUMP IF TOKEN IS RW
071D 12	4011	STAX	D	PUT CHARACTER IN BUFFER
071E FE 0D	4012	CPI	CR	CHECK FOR DONE
0720 C8	4013	RZ		
0721 13	4014	INX	D	ADVANCE DESTINATION BUFFER ADDRESS
0722 C3 17 07	4015	JMP	UPPO	
0725	4016	* COME HERE WHEN RW BYTE DETECTED IN SOURCE		
0725 E5	4017 UPP1	PUSH	H	SAVE SOURCE POINTER
0726 24 02 08	4018	LXI	H,RWT	BASE OF RWT
0729 BE	4019 UPP2	CMP	M	SEE IF RW MATCHED RWT ENTRY
072A 23	4020	INX	H	ADVANCE RWT POINTER
072B C2 29 07	4021	JNZ	UPP2	CONTINUE LOOKING IF NOT FOUND
072E	4022	* FOUND MATCH, ENTRY POINTER LOCATES FIRST CHARACTER		
072E 7E	4023 UPP3	MOV	A,M	CHARACTER OF RW
072F B7	4024	ORA	A	CHECK FOR DONE
0730 FA 39 07	4025	JM	UPP4	
0733 12	4026	STAX	D	
0734 13	4027	INX	D	
0735 23	4028	INX	H	
0736 C3 2E 07	4029	JMP	UPP3	
0739	4030	* COME HERE IF DONE WITH RW TRANSFER		
0739 E1	4031 UPP4	POP	H	SOURCE POINTER
073A C3 17 07	4032	JMP	UPPO	
073D	4033	*		
073D	4034	CONSTANTS AND TABLES		
073D	4035	*		
073D 52 45 41 44	4036 RDYS	ASC		'READY''
59 22				
0743 50 52 4F 47	4037 PLS	ASC		'PROGRAM LOADED? ''
52 41 4D 20				
4C 4F 41 44				
45 44 3F 20				
22				
0754 20 45 52 52	4038 ERS	ASC		' ERROR''
4F 52 22				
075B 20 49 4E 20	4039 INS	ASC		' IN LINE ''
4C 49 4E 45				
20 22				
0765 53 54 4F 50	4040 STOPS	ASC		'STOP''
22				
076A 46 49 52 53	4041 FAS	ASC		'FIRST ADDR ''
54 20 41 44				
44 52 20 22				
0776 4C 41 53 54	4042 LAS	ASC		'LAST ADDR ''
20 41 44 44				
52 20 22				
0781	4043	*		
0781 FF	4044	DB	-1	FLAGS END OF SINE COEFFICIENT LIST
0782 00	4045	DB	0	
0783 10	4046	DB	1*16	
0784 00 00	4047	DW	0	
0786 00	4048	DB	0	
0787 81	4049 FPONE	DB	129	EXPONENT

0788	1050	*	SINE COEFFICIENT LIST	
0788	1051	*	NOTE: THE FLOATING PNT 4 ABOVE IS A PART OF THIS TABLE	
0788 16	1052	DB	1*16+6	
0789 66	1053	DB	6*16+6	
078A 67	1054	DB	6*16+7	
078B 01	1055	DB	1	
078C 80	1056	DB	128	-.166667 E 0 (-1/31)
078D 83	1057	DB	8*16+3	
078E 33	1058	DB	3*16+3	
078F 33	1059	DB	3*16+3	
0790 00	1060	DB	0	
0791 7E	1061	DB	128-2	.833333 E-2 (1/51)
0792 49	1062	DB	1*16+9	
0793 84	1063	DB	8*16+4	
0794 13	1064	DB	1*16+3	
0795 01	1065	DB	1	
0796 7D	1066	DB	128-3	-.198413 E-3 (-1/71)
0797 27	1067	DB	2*16+7	
0798 55	1068	DB	5*16+5	
0799 73	1069	DB	7*16+3	
079A 00	1070	DB	0	
079B 7B	1071	DB	128-5	.275573 E-5 (1/91)
079C 25	1072	DB	2*16+5	
079D 05	1073	DB	0*16+5	
079E 24	1074	DB	2*16+1	
079F 04	1075	DB	4	
07A0 79	1076	SINX DB	128-7	-.250521 E-7 (-1/111)
07A1	1077	*	COSINE COEFFICIENT LIST	
07A1 FF	1078	DB	-1	MARKS END OF LIST
07A2 00	1079	DB	0	
07A3 10	1080	DB	1*16+0	
07A4 00	1081	DB	0	
07A5 00	1082	DB	0	
07A6 00	1083	DB	0	
07A7 81	1084	DB	128+1	.100000 E 1 (1/1)
07A8 50	1085	DB	5*16+0	
07A9 00	1086	DB	0	
07AA 00	1087	DB	0	
07AB 01	1088	DB	1	
07AC 80	1089	MATUB DB	128	-.500000 E 0 (-1/21)
07AD 41	1090	DB	4*16+1	
07AE 66	1091	DB	6*16+6	
07AF 67	1092	DB	6*16+7	
07B0 00	1093	DB	0	
07B1 7F	1094	RANDS DB	128-1	.416667 E-1 (1/41)
07B2 13	1095	DB	1*16+3	
07B3 88	1096	DB	8*16+8	
07B4 89	1097	DB	8*16+9	
07B5 01	1098	DB	1	
07B6 7E	1099	DB	128-2	.138889 E-2 (-1/61)
07B7 24	1100	DB	2*16+4	
07B8 80	1101	DB	8*16+0	
07B9 16	1102	DB	1*16+6	
07BA 00	1103	DB	0	
07BB 7C	1104	DB	128-4	.248016 E-4 (1/81)
07BC 27	1105	DB	2*16+7	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

07BD	55		1106	DB	5*16+5	
07BE	73		1107	DB	7*16+3	
07BF	01		1108	DB	1	
07C0	7A		1109	COSX	128-6	.275573 E-6 (-1/101)
07C1	20		1110	DB	2*16	
07C2	00	00	1111	DW	0	
07C4	00		1112	DB	0	
07C5	81		1113	FPTWO	129	
07C6	15		1114	DB	1*16+5	
07C7	70		1115	DB	7*16+0	
07C8	80		1116	DB	8*16+0	
07C9	00		1117	DB	0	
07CA	81		1118	PIC2	128+1	PI/2 .157080 E 4
07CB	63		1119	DB	6*16+3	
07CC	66		1120	DB	6*16+6	
07CD	20		1121	DB	2*16+0	
07CE	00		1122	DB	0	
07CF	80		1123	PIC1	128	2/PI .636620 E 0
07D0	97	17	1124	LCSTKA	DW	CSTKL
07D2			1125	*		
07D2			1126	*		
07D2			1127	*		
07D2	55	02	1128	CMNDD	DW	CRUN 0
07D4	15	02	1129		DW	CLIST 1
07D6	08	02	1130		DW	CNULL 2
07D8	17	01	1131		DW	CSCR 3
07DA	00	00	1132		DW	START 4 SET UP MEMORY BOUNDS
07DC	00	00	1133		DW	TSAV 5 TAPE SAVE JUMP ADDRESS POINTER
07DE	00	00	1134		DW	TLOAD 6 TAPE LOAD
07E0			1135	*		
07E0	9C	02	1136	STATD	DW	LET 0
07E2	1B	03	1137		DW	NEXT 1
07E4	81	03	1138		DW	SIF 2
07E6	A0	03	1139		DW	SGOTO 3
07E8	B4	03	1140		DW	GOSUB 4
07EA	DA	03	1141		DW	RETRN 5
07EC	5E	04	1142		DW	READ 6
07EE	05	04	1143		DW	DATA 7
07F0	B0	02	1144		DW	SFOR 10
07F2	CC	04	1145		DW	PRINT 11
07F4	7A	05	1146		DW	INPUT 12
07F6	15	04	1147		DW	DIM 13
07F8	4F	04	1148		DW	STOP 14
07FA	62	00	1149		DW	END 15
07FC	C2	04	1150		DW	RESTOR 16
07FE	08	04	1151		DW	REM 17
0800	DF	01	1152		DW	CCLEAR 20
0802			1153	*		
0802			1154	*		R/W WORD TABLE FORMAT IS RESERVED WORD FOLLOWED BY CHR
0802			1155	*		OF RESERVED WORD. LAST ENTRY IS FOLLOWED BY A 377Q
0802			1156	*		RW'S THAT ARE SUBSTRINGS OF OTHER RW'S (E.G. >) MUST
0802			1157	*		FOLLOW THE LARGER WORD.
0802			1158	*		
0802	80		1159	RWT	DB	200Q
0803	4C	45 54	1160		ASC	"LET"
0806	81		1161		DB	201Q

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0807	4E 45 58 54	1162	ASC	"NEXT"
080B	82	1163	DB	202Q
080C	49 46	1164	ASC	"IF"
080E	83	1165	DB	203Q
080F	47 4F 54 4F	1166	ASC	"GOTO"
0813	84	1167	DB	204Q
0814	47 4F 53 55	1168	ASC	"GOSUB"
	42			
0819	85	1169	DB	205Q
081A	52 45 54 55	1170	ASC	"RETURN"
	52 4E			
0820	86	1171	DB	206Q
0821	52 45 41 44	1172	ASC	"READ"
0825	87	1173	DB	207Q
0826	44 41 54 41	1174	ASC	'DATA'
082A		1175	EQU	207Q
082A	88	1176	DB	210Q
082B	46 4F 52	1177	ASC	"FOR"
082E	89	1178	DB	211Q
082F	50 52 49 4E	1179	ASC	"PRINT"
	54			
0834	89	1180	DB	211Q
0835	3A	1181	DB	':'
0836	8A	1182	DB	212Q
0837	49 4E 50 55	1183	ASC	"INPUT"
	54			
083C	8B	1184	DB	213Q
083D	44 49 4D	1185	ASC	'DIM'
0840	8C	1186	DB	214Q
0841	53 54 4F 50	1187	ASC	'STOP'
0845	8D	1188	DB	215Q
0846	45 4E 44	1189	ASC	'END'
0849	8E	1190	DB	216Q
084A	52 45 53 54	1191	ASC	'RESTORE'
	4F 52 45			
0851	8F	1192	DB	217Q
0852	52 45 4D	1193	ASC	'REM'
0855	90	1194	DB	220Q
0856	43 4C 45 43	1195	ASC	'CLEAR'
	52			
085B		1196	EQU	220Q
085B		1197	EQU	224Q
				LAST INITIAL RESERVED WORD VALUE + 1
085B	9F	1198	DB	237Q
085C	53 54 45 50	1199	ASC	'STEP'
0860		1200	EQU	237Q
0860	9E ..	1201	DB	236Q
0861	54 4F	1202	ASC	'TO'
0863		1203	EQU	236Q
0863	9D	1204	DB	235Q
0864	54 48 45 4E	1205	ASC	'THEN'
0868		1206	EQU	235Q
0868	9C	1207	DB	234Q
0869	54 43 42	1208	ASC	'TAB'
086C		1209	EQU	234Q
086C	A0	1210	DB	240Q
086D	52 55 4E	1211	ASC	'RUN'

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0870		1212	RUNRW	EQU	240Q
0870	A1	1213		DB	241Q
0871	4C 49 53 54	1214		ASC	'LIST'
0875		1215	LISTRW	EQU	241Q
0875	A2	1216		DB	242Q
0876	4E 55 4C 4C	1217		ASC	'NULL'
087A		1218	NULLRW	EQU	242Q
087A	A3	1219		DB	243Q
087B	53 43 52	1220		ASC	'SCR'
087E		1221	SCRRW	EQU	243Q
087E	A4	1222		DB	244Q
087F	4D 45 4D	1223		ASC	'MEM'
0882		1224	MEMRW	EQU	245Q
0882	A5	1225		DB	245Q
0883	54 53 41 56	1226		ASC	"TSAV"
0887	A6	1227		DB	246Q
0888	54 4C 4F 41	1228		ASC	"TLOAD"
	44				
088D		1229	LPARRW	EQU	'(' -OPBASE+340Q
088D	E0	1230		DB	LPARRW
088E	28	1231		DB	'('
088F	E2	1232		DB	'*' -OPBASE+340Q
0890	2A	1233		DB	'*'
0891		1234	PLSRW	EQU	'+' -OPBASE+340Q
0891	E3	1235		DB	PLSRW
0892	2B	1236		DB	'+'
0893		1237	MINRW	EQU	'-' -OPBASE+340Q
0893	E5	1238		DB	MINRW
0894	2D	1239		DB	'-'
0895	E7	1240		DB	'/' -OPBASE+340Q
0896	2F	1241		DB	'/'
0897	EF	1242		DB	67Q-OPBASE+340Q
0898	3E 3D	1243		ASC	'>='
089A	F0	1244		DB	70Q-OPBASE+340Q
089B	3C 3D	1245		ASC	'<='
089D	F1	1246		DB	71Q-OPBASE+340Q
089E	3C 3E	1247		ASC	'<>'
08A0	EA	1248		DB	62Q-OPBASE+340Q
08A1	3D 3E	1249		ASC	'=>'
08A3	EB	1250		DB	63Q-OPBASE+340Q
08A4	3D 3C	1251		ASC	'=<'
08A6	F4	1252		DB	'<' -OPBASE+340Q
08A7	3C	1253		DB	'<'
08A8		1254	EQRW	EQU	'=' -OPBASE+340Q
08A8	F5	1255		DB	EQRW
08A9	3D	1256		DB	'='
08AA	F6	1257		DB	'>' -OPBASE+340Q
08AB	3E	1258		DB	'>'
08AC	C1	1259		DB	304Q
08AD	41 42 53	1260		ASC	'ABS'
08B0	C6	1261		DB	306Q
08B1	49 4E 54	1262		ASC	'INT'
08B4	CC	1263		DB	314Q
08B5	41 52 47	1264		ASC	'ARG'
08B8	CD	1265		DB	315Q
08B9	43 41 4C 4C	1266		ASC	'CALL'

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

08BD CE	1267	DB	316Q
08BE 52 4E 44	1268	ASC	'RND'
08C1 D2	1269	DB	322Q
08C2 53 47 4E	1270	ASC	'SGN'
08C5 D3	1271	DB	323Q
08C6 53 49 4E	1272	ASC	'SIN'
08C9 C4	1273	DB	304Q
08CA 53 51 52	1274	ASC	'SQR'
08CD D7	1275	DB	327Q
08CE 54 41 4E	1276	ASC	'TAN'
08D1 D8	1277	DB	330Q
08D2 43 4F 53	1278	ASC	'COS'
08D5 FF	1279	DB	377Q
08D6	1280 *		
08D6	1281 *		
08D6	1282 *		
08D6 OF	1283 OPTAB	DB	15
08D7	1284 OPLPAR	EQU	OPTAB
08D7 BC 09	1285	DW	ALPAR
08D9 OF	1286	DB	15
08DA CC 09	1287	DW	AABS
08DC OA	1288	DB	10.
08DD 9D 09	1289	DW	AMUL
08DF 06	1290	DB	6
08E0 89 09	1291	DW	AADD
08E2 OF	1292	DB	15
08E3 57 OA	1293	DW	ASQR
08E5 06	1294	DB	6
08E6 93 09	1295	DW	ASUB
08E8 OF	1296	DB	15
08E9 AA 0B	1297	DW	AINT
08EB OA	1298	DB	40
08EC A7 09	1299	DW	ADIV
08EE 01	1300 OPBOL	DB	1
08EF 00 00	1301	DW	0
08F1 0D	1302	DB	13
08F2 BD 09	1303	DW	ANEG
08F4 04	1304	DB	4
08F5 4C 09	1305	DW	AGE
08F7 04	1306	DB	4
08F8 58 09	1307	DW	ALE
08FA OF	1308	DB	15
08FB 79 0B	1309	DW	AARG
08FD OF	1310	DB	15
08FE 86 0B	1311	DW	ACALL
0900 OF ..	1312	DB	15
0901 AD OA	1313	DW	ARND
0903 04	1314	DB	4
0904 4C 09	1315	DW	AGE
0906 04	1316	DB	4
0907 58 09	1317	DW	ALE
0909 04	1318	DB	4
090A 43 09	1319	DW	ANE
090C OF	1320	DB	15
090D D2 09	1321	DW	ASGN
090F OF	1322	DB	15

OPERATION TABLE

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0910	ED	09	1323	DW	ASIN	
0912	04		1324	DB	4	
0913	2D	09	1325	DW	ALT	
0915	04		1326	DB	4	
0916	3A	09	1327	DW	AEQ	
0918	04		1328	DB	4	
0919	21	09	1329	DW	AGT	
091B	0F		1330	DB	15	
091C	3C	0A	1331	DW	ATAN	
091E	0F		1332	DB	15	
091F	30	0A	1333	DW	ACOS	
0921			1334	*		
0921			1335	*		
0921			1336	*		
0921	CD	64 09	1337	AGT	CALL	RELOP
0924	CA	2A 09	1338		JZ	RFALSE
0927	FA	36 09	1339		JM	RTRUE
092A	AF		1340	RFALSE	XRA	A
092B	12		1341		STAX	D
092C	C9		1342		RET	
092D	CD	64 09	1343	ALT	CALL	RELOP
0930	CA	2A 09	1344		JZ	RFALSE
0933	FA	2A 09	1345		JM	RFALSE
0936	3E	FF	1346	RTRUE	MVI	A,377Q
0938	12		1347		STAX	D
0939	C9		1348		RET	
093A	CD	64 09	1349	AEQ	CALL	RELOP
093D	CA	36 09	1350		JZ	RTRUE
0940	C3	2A 09	1351		JMP	RFALSE
0943	CD	64 09	1352	ANE	CALL	RELOP
0946	CA	2A 09	1353		JZ	RFALSE
0949	C3	36 09	1354		JMP	RTRUE
094C	CD	64 09	1355	AGE	CALL	RELOP
094F	CA	36 09	1356		JZ	RTRUE
0952	FA	36 09	1357		JM	RTRUE
0955	C3	2A 09	1358		JMP	RFALSE
0958	CD	64 09	1359	ALE	CALL	RELOP
095B	CA	36 09	1360		JZ	RTRUE
095E	FA	2A 09	1361		JM	RFALSE
0961	C3	36 09	1362		JMP	RTRUE
0964			1363	*		COMMON ROUTINE FOR RELATIONAL OPERATOR ACTION
0964			1364	*		LEFT ARG ADDR IN DE, SAVED
0964			1365	*		RIGHT ARG ADDR IN BC
0964			1366	*		ON RETURN SIGN SET=GT, ZERO SET=EQUAL
0964	D5		1367	RELOP	PUSH	D
0965	0B		1368		DCX	B
0966	1B		1369		DCX	D
0967	60		1370		MOV	H,B
0968	69		1371		MOV	L,C
0969	1A		1372		LDAX	D
096A	96		1373		SUB	M
096B	23		1374		INX	H
096C	13		1375		INX	D
096D	C2	82 09	1376		JNZ	RLOP4 TEST SIGNS OF ARGS IF DIFFERENT THEN RET
0970	01	BB 18	1377		LXI	B,FPSINK
0973	CD	1F 14	1378		CALL	FSUB

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0976	3A	BB	18	1379	LDA	FPSINK	CHECK FOR ZERO RESULT
0979	B7			1380	ORA	A	
097A	CA	82	09	1381	JZ	RLOP1	
097D	3A	BA	18	1382	LDA	FPSINK-1	SIGN OF FPSINK
0980	07			1383	RLC		
0981	3D			1384	DCR	A	
0982	3E	01		1385	RLOP1	MVI	A,1
0984	32	91	17	1386	STA	RELTYP	SET RELTYP TRUE
0987	D1			1387	POP	D	
0988	C9			1388	RET		
0989				1389	*		
0989				1390	*	ACTION ROUTINES FOR ARITHMETIC OPERATORS	
0989				1391	*	(CODE WASTERS)	
0989	60			1392	AADD	MOV	H,B
098A	69			1393		MOV	L,C
098B	42			1394		MOV	B,D
098C	4B			1395		MOV	C,E
098D	CD	C8	13	1396	AADD1	CALL	FADD
0990	C3	AE	09	1397		JMP	FPETST
0993	60			1398	ASUB	MOV	H,B
0994	69			1399		MOV	L,C
0995	42			1400		MOV	B,D
0996	4B			1401		MOV	C,E
0997	CD	1F	14	1402	ASUB1	CALL	FSUB
099A	C3	AE	09	1403		JMP	FPETST
099D	60			1404	AMUL	MOV	H,B
099E	69			1405		MOV	L,C
099F	42			1406		MOV	B,D
09A0	4B			1407		MOV	C,E
09A1	CD	98	14	1408	AMUL1	CALL	FMUL
09A4	C3	AE	09	1409		JMP	FPETST
09A7	60			1413	ADIV	MOV	H,B
09A8	69			1414		MOV	L,C
09A9	42			1415		MOV	B,D
09AA	4B			1416		MOV	C,E
09AB	CD	A1	15	1417	ADIV1	CALL	FDIV
09AE	AF			1418	FPETST	XRA	A
09AF	32	91	17	1419		STA	RELTYP
09B2	3A	23	17	1420		LDA	ERRI
09B5	B7			1421		ORA	A
09B6	C8			1422		RZ	
09B7	2A	1C	19	1423		LHLD	ASTKA
09BA	36	00		1424	FPET1	MVI	M,0
09BC	C9			1425	ALPAR	RET	
09BD				1426	*		
09BD				1427	*	UNARY AND BUILT IN FUNCTION ACTION ROUTINES	
09BD				1428	*		
09BD	0A			1429	ANEG	LDAX	B
09BE	B7			1430		ORA	A
09BF	CA	C7	09	1431		JZ	ANEG1
09C2	0B			1432		DCX	B
09C3	0A			1433		LDAX	B
09C4	EE	01		1434		XRI	1
09C6	02			1435		STAX	B
09C7	AF			1436	ANEG1	XRA	A
09C8	32	91	17	1437		STA	RELTYP

09CB	C9	1438	RET		
09CC	0B	1439	AABS	DCX	B
09CD	AF	1440		XRA	A
09CE	02	1441		STAX	B
09CF	C3 C7 09	1442		JMP	ANEG1
09D2	CD C7 09	1443	ASGN	CALL	ANEG1
09D5	50	1444		MOV	D,B
09D6	59	1445		MOV	E,C
09D7	0A	1446		LDAX	B
09D8	B7	1447		ORA	A
09D9	C2 DE 09	1448		JNZ	ASGN1
09DC	12	1449		STAX	D
09DD	C9	1450		RET	
09DE	0B	1451	ASGN1	DCX	B
09DF	0A	1452		LDAX	B
09E0	B7	1453		ORA	A
09E1	21 87 07	1454		LXI	H,FPONE
09E4	CA 9C 0C	1455		JZ	VCOPY
09E7	21 C7 13	1456		LXI	H,FPNONE
09EA	C3 9C 0C	1457		JMP	VCOPY
09ED		1458	*		
09ED		1459	*	COMPUTE SIN(X) X=TOP OF ARGUMENT STACK	
09ED		1460	*	RETURN RESULT IN PLACE OF X	
09ED		1461	*		
09ED	CD 31 0B	1462	ASIN	CALL	QUADC
09F0	2A 1C 19	1463		LHLD	ASTKA
09F3	54	1464		MOV	D,H
09F4	5D	1465		MOV	E,L
09F5	01 C0 18	1466		LXI	B,FTEMP
09F8	CD A1 09	1467		CALL	AMUL1
09FB	F1	1468		POP	PSW
09FC	F5	1469		PUSH	PSW
09FD	1F	1470		RAR	
09FE	DA 27 0A	1471		JC	SIN10
0A01		1472	*	COMPUTE X*P(X*X) -- SINE	
0A01	11 C5 18	1473		LXI	D,FTEM1
0A04	2A 1C 19	1474		LHLD	ASTKA
0A07	CD 9C 0C	1475		CALL	VCOPY
0A0A	01 A0 07	1476		LXI	B,SINX
0A0D	CD 03 0B	1477		CALL	POLY
0A10	CD 2A 0B	1478		CALL	PREPOP
0A13	21 C5 18	1479		LXI	H,FTEM1
0A16	CD A1 09	1480		CALL	AMUL1
0A19		1481	*	COMPUTE SIGN OF RESULT	
0A19		1482	*	POSITIVE FOR QUADRANTS 0,1. NEGATIVE FOR 2,3	
0A19		1483	*	NEGATE ABOVE FOR NEGATIVE ARGUMENTS	
0A19	F1	1484	SIN5	POP	PSW
0A1A	47	1485		MOV	B,A
0A1B	F1	1486		POP	PSW
0A1C	07	1487		RLC	.
0A1D	A8	1488		XRA	B
0A1E	2A 1C 19	1489		LHLD	ASTKA
0A21	2B	1490		DCX	H
0A22	D6 02	1491		SUI	2
0A24	F8	1492		RM	.
0A25	34	1493		INR	M

GET EXPONENT

MAKE ARGUMENT ZERO

COMPUTE QUADRANT

FTEMP=X*X

A=QUADRANT

QUAD ODD. COMPUTE COSINE

FTEM1=X*X

P(X*X)

X*P(X*X)

SIGN

SIGN, 2 TO THE 4ST BIT

QUADRANT, MAYBE MODIFIED FOR NEGATIVE ARG.

PTR TO SIGN

QUADRANT 0 OR 1

ELSE SET RESULT NEGATIVE

0A26 C9	1494	RET		
0A27	1495	* COMPUTE P(X*X) -- COSINE		
0A27 01 C0 07	1496	SIN10 LXI	B,COSX	
0A2A CD 03 0B	1497	CALL	POLY	P(X*X)
0A2D C3 19 0A	1498	JMP	SIN5	
0A30	1499	*		
0A30	1500	* COMPUTE COS(X) X=TOP OF ARGUMENT STACK		
0A30	1501	* RETURN RESULT IN PLACE OF X		
0A30	1502	* COS(X)=SIN(X+PI/2)		
0A30	1503	*		
0A30 CD 2A 0B	1504	ACOS CALL	PREPOP	
0A33 24 CA 07	1505	LXI	H,PIC2	PI/2
0A36 CD 8D 09	1506	CALL	AADD1	TOS=TOS+PI/2
0A39 C3 ED 09	1507	JMP	ASIN	
0A3C	1508	*		
0A3C	1509	* COMPUTE TAN(X) X=TOP OF ARGUMENT STACK		
0A3C	1510	* RETURN RESULT IN PLACE OF X		
0A3C	1511	* TAN(X)=SIN(X)/COS(X)		
0A3C	1512	*		
0A3C 2A 1C 19	1513	ATAN LHL	ASTKA	
0A3F CD A7 0C	1514	CALL	PSHAS	PUSH COPY OF X ONTO ARG STACK
0A42 CD 30 0A	1515	CALL	ACOS	COS(X)
0A45 11 CA 18	1516	LXI	D,FTEM2	
0A48 CD BE 0C	1517	CALL	POPA1	FTEM2=COS(X)
0A4B CD ED 09	1518	CALL	ASIN	
0A4E CD 2A 0B	1519	CALL	PREPOP	
0A51 21 CA 18	1520	LXI	H,FTEM2	
0A54 C3 AB 09	1521	JMP	ADIV1	SIN(X)/COS(X)
0A57	1522	*		
0A57	1523	* COMPUTE SQR(X) X=TOP OF ARGUMENT STACK		
0A57	1524	* RETURN RESULT IN PLACE OF X		
0A57	1525	*		
0A57 2A 1C 19	1526	ASQR LHL	ASTKA	
0A5A 11 C0 18	1527	LXI	D,FTEMP	
0A5D CD 9C 0C	1528	CALL	VCOPY	SAVE X IN FTEMP
0A60	1529	* COMPUTE EXPONENT OF FIRST GUESS AS EXPONENT OF X/2		
0A60 2A 1C 19	1530	LHL	ASTKA	
0A63 7E	1531	MOV	A,M	
0A64 B7	1532	ORA	A	
0A65 C8	1533	RZ	.	X=0
0A66 D6 80	1534	SUI	128	
0A68 FA 71 0A	1535	JM	SQR5	NEGATIVE EXPONENT
0A6B 0F	1536	RRC		
0A6C E6 7F	1537	ANI	127	
0A6E C3 78 0A	1538	JMP	SQR6	
0A71 2F	1539	SQR5 CMA		
0A72 3C	1540	INR	A	
0A73 0F	1541	RRC		
0A74 E6 7F	1542	ANI	127	
0A76 2F	1543	CMA		
0A77 3C	1544	INR	A	
0A78 C6 80	1545	SQR6 ADI	128	
0A7A 77	1546	MOV	M,A	
0A7B	1547	* TEST FOR NEGATIVE ARGUMENT		
0A7B 2B	1548	DCX	H	
0A7C 7E	1549	MOV	A,M	

0A7D 01 41 4E	1550	LXI	B,'NA'	
0A80 B7	1551	ORA	A	
0A81 C2 C5 00	1552	JNZ	ERROR	NEG ARGUMENT
0A84	1553	* DO NEWTON ITERATIONS		
0A84	1554	* NEWGUESS =(X/OLDGUESS + OLDGUESS) /2		
0A84 3E 06	1555	MVI	A,6	DO 6 ITERATIONS
0A86 F5	1556	SQR20	PUSH	PSW
0A87 01 C5 18	1557	LXI	B,FTEM1	SET NEW ITERATION COUNT
0A8A 11 C0 18	1558	LXI	D,FTEMP	FTEMP IS 'X'
0A8D 2A 1C 19	1559	LHLD	ASTKA	GUESS
0A90 CD AB 09	1560	CALL	ADIV1	FTEM1=X/GUESS
0A93 11 C5 18	1561	LXI	D,FTEM1	
0A96 2A 1C 19	1562	LHLD	ASTKA	
0A99 44	1563	MOV	B,H	
0A9A 4D	1564	MOV	C,L	
0A9B CD 8D 09	1565	CALL	AADD1	TOS=(X/GUESS)+GUESS
0A9E CD 2A 0B	1566	CALL	PREPOP	
0AA1 24 C5 07	1567	LXI	H,FPTWO	
0AA4 CD AB 09	1568	CALL	ADIV1	TOS= (X/GUESS+GUESS)/2
0AA7 F4	1569	POP	PSW	
0AA8 3D	1570	DCR	A	DECREMENT COUNT
0AA9 C2 86 0A	1571	JNZ	SQR20	DO ANOTHER ITERATION
0AAC C9	1572	RET		
0AAD	1573	*		
0AAD	1574	* COMPUTE RND(X) X=TOP OF ARGUMENT STACK		
0AAD	1575	* FRAND IS UPDATED TO NEW RANDOM VALUE		
0AAD	1576	* A RANDOM NUMBER IN THE RANGE 0<RND<1 IS RETURNED IN PLACE		
0AAD	1577	*		
0AAD CD 2A 0B	1578	ARND	CALL	PREPOP
0AB0 11 CF 18	1579	LXI	D,FRAND	
0AB3 21 CF 18	1580	LXI	H,FRAND	
0AB6 CD A1 09	1581	CALL	AMUL1	TOS=FRAND*FRAND
0AB9	1582	* SET EXPONENT=0		
0AB9 2A 1C 19	1583	LHLD	ASTKA	
0ABC 36 80	1584	MVI	M,128	EXPONENT=128 (0 IN EXTERNAL FORM)
0ABE	1585	* PERMUTE DIGITS OF X AS		
0ABE	1586	* 123456 INTO 345612		
0ABE 01 FC FF	1587	LXI	B,-4	
0AC1 09	1588	DAD	B	
0AC2 46	1589	MOV	B,M	SAVE 12
0AC3 23	1590	INX	H	
0AC4 23	1591	INX	H	
0AC5 CD FE 0A	1592	CALL	PERMU	56=12
0AC8 CD FE 0A	1593	CALL	PERMU	34=56
0ACB CD FE 0A	1594	CALL	PERMU	12=34
0ACE	1595	* NORMALIZE NUMBER		
0ACE 2A 1C 19	1596	RND5	LHLD	ASTKA
0AD1 01 FC FF	1597	LXI	B,-FPSIZ+1	TOS
0AD4 09	1598	DAD	B	
0AD5 7E	1599	MOV	A,M	FIRST DIGIT PAIR
0AD6 E6 F0	1600	ANI	15*16	
0AD8 C2 F4 0A	1601	JNZ	RND10	NUMBER IS NORMALIZED
0ADB	1602	* SHIFT LEFT ONE DIGIT		
0ADB 2A 1C 19	1603	LHLD	ASTKA	
0ADE 7E	1604	MOV	A,M	EXPONENT
0ADF 3D	1605	DCR	A	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0AEO	32 29 17	1606	STA	EXP	
0AE3	CD 84 16	1607	CALL	LOAD	TOS INTO TEMP
0AE6	06 04	1608	MVI	B,4	
0AE8	CD CE 16	1609	CALL	LEFT	SHIFT LEFT
0AEB	CD 2A 0B	1610	CALL	PREPOP	
0AEE	CD 9A 16	1611	CALL	STORE	
0AF4	C3 CE 0A	1612	JMP	RND5	TEST IF NORMALIZED
0AF4		1613	* SAVE NEW RANDM NUMBER IN FRAND CELL		
0AF4	11 CF 18	1614	RND10	LXI	D,FRAND
0AF7	2A 1C 19	1615	LHLD	ASTKA	
0AFA	CD 9C 0C	1616	CALL	VCOPY	FRAND=TOS
0AFD	C9	1617	RET		
0AFE		1618	* PERMUTE PAIR OF DIGIT PAIRS		
0AFE	7E	1619	PERMU	MOV	A,M
0AFF	70	1620	MOV	M,B	
0B00	47	1621	MOV	B,A	
0B01	2B	1622	DCX	H	
0B02	C9	1623	RET		
0B03		1624	*		
0B03		1625	* EVALUATE P(X) USING HORNERS METHOD (X IS IN FTEMP)		
0B03		1626	* COEFFICIENT LIST POINTER IS IN BC		
0B03		1627	* RESULT REPLACES NUMBER ON TOP OF ARGUMENT STACK (Y)		
0B03	2A 1C 19	1628	POLY	LHLD	ASTKA
0B06	EB	1629	XCHG	.	DE=PTR TO Y
0B07	60	1630	MOV	H,B	
0B08	69	1631	MOV	L,C	HL PTR TO COEFFICIENT LIST
0B09	CD 9C 0C	1632	CALL	VCOPY	Y=FIRST COEFFICIENT
0B0C		1633	* MULTIPLY BY X		
0B0C	E5	1634	POLY1	PUSH	H
0B0D	CD 2A 0B	1635	CALL	PREPOP	SAVE COEFF LIST POINTER
0B10	21 C0 18	1636	LXI	H,FTEMP	
0B13	CD A4 09	1637	CALL	AMUL1	Y=Y*X
0B16		1638	* ADD NEXT COEFF		
0B16	CD 2A 0B	1639	CALL	PREPOP	
0B19	E4	1640	POP	H	
0B1A	E5	1641	PUSH	H	HL=COEFF. LIST POINTER
0B1B	CD 8D 09	1642	CALL	AADD1	Y=Y+COEFF.
0B1E		1643	* BUMP POINTER TO NEXT COEFFICIENT		
0B1E	E4	1644	POP	H	COEFF. POINTER
0B1F	01 FA FF	1645	LXI	B,-FPSIZ-1	
0B22	09	1646	DAD	B	NEXT COEF SIGN
0B23	7E	1647	MOV	A,M	
0B24	23	1648	INX	H	PTR TO EXPONENT
0B25	B7	1649	ORA	A	
0B26	F2 0C 0B	1650	JP	POLY1	PROCESS NEXT COEFFICIENT
0B29	C9	1651	RET	.	NEGATIVE SIGN (-1) - ENDS LIST
0B2A		1652	*		
0B2A		1653	* PREPARE FOR OPERATION		
0B2A		1654	*		
0B2A	2A 1C 19	1655	PREPOP	LHLD	ASTKA
0B2D	EB	1656	XCHG	.	DE=ASTKA
0B2E	42	1657	MOV	B,D	
0B2F	4B	1658	MOV	C,E	
0B30	C9	1659	RET		
0B31		1660	*		
0B31		1661	* QUADRANT COMPUTATION		

```

OB34 1662 * POPS TOP OF ARGUMENT STACK
OB34 1663 * COMPUTE/GETS SIGN OF ARGUMENT, QUADRANT OF ARGUMENT
OB34 1664 * AND INDEX INTO QUADRANT
OB34 1665 *
OB34 1666 *     EXITS WITH:
OB34 1667 * SP POINTING TO QUADRANT, MOD 4
OB34 1668 * SP+2 POINTING TO SIGN OF ARGUMENT
OB34 1669 * TOP OF ARGUMENT STACK HAS INDEX INTO QUADRANT
OB34 2A 1C 19 1670 QUADC LHLD ASTKA
OB34 2B 1671 DCX H POINT TO SIGN
OB35 46 1672 MOV B,M
OB36 AF 1673 XRA A
OB37 77 1674 MOV M,A ARG. SIGN=0
OB38 60 1675 MOV H,B
OB39 E3 1676 XTHL . PUT SIGN ON STACK, POP RETURN
OB3A E5 1677 PUSH H PUSH RETURN
OB3B 1678 * COMPUTE QUADRANT OF ABS(X)
OB3B 2A 1C 19 1679 LHLD ASTKA
OB3E CD A7 OC 1680 CALL PSHAS PUT COPY OF ARG. ONTO STACK
OB41 CD 2A OB 1681 CALL PREPOP
OB44 24 CF 07 1682 LXI H,PIC4 2/PI
OB47 CD A4 09 1683 CALL AMUL4 TOS=X*2/PI
OB4A CD 2A OB 1684 CALL PREPOP
OB4D CD AA OB 1685 CALL AINT TOS=INT(X*2/PI)
OB50 2A 1C 19 1686 LHLD ASTKA
OB53 CD A7 OC 1687 CALL PSHAS ANOTHER COPY
OB56 CD 8A OD 1688 CALL PFIX POP TOS TO DE
OB59 7B 1689 MOV A,E
OB5A F5 1690 PUSH PSW QUADRANT
OB5B CD 2A OB 1691 CALL PREPOP
OB5E 24 CA 07 1692 LXI H,PIC2
OB61 CD A4 09 1693 CALL AMUL4 TOS=INT(X*2/PI)
OB64 14 CO 18 1694 LXI D,FTEMP
OB67 CD BE OC 1695 CALL POPA4 FTEMP=TOS
OB6A CD 2A OB 1696 CALL PREPOP
OB6D 24 CO 18 1697 LXI H,FTEMP
OB70 CD 97 09 1698 CALL ASUB4 TOS=TOS-FTEMP
OB73 F4 1699 POP PSW A=QUADRANT, LOW ORDER BYTE
OB74 E6 03 1700 ANI 3 MOD 4
OB76 E4 1701 POP H
OB77 F5 1702 PUSH PSW SAVE QUADRANT ON STACK
OB78 E9 1703 PCHL . RETURN
OB79 1704 * SET UP ARG FOR USER CALL
OB79 CD 8A OD 1705 AARG CALL PFIX
OB7C EB 1706 XCHG
OB7D 22 4C 19 1707 SHLD CALLA
OB80 14 BB 18 1708 LXI D,FPSIN
OB83 C3 A8 OC 1709 JMP PSHA3 PUTS BACK THE ARG VALUE ON ARG STACK
OB86 1710 * USED TO CALL USER ROUTINE
OB86 CD 8A OD 1711 ACALL CALL PFIX GET THE ADDRESS
OB89 2A 4C 19 1712 LHLD CALLA GET THE USER ARGUMENT
OB8C EB 1713 XCHG
OB8D 01 92 OB 1714 LXI B,ACAL4 RETURN LINK FOR USER ROUTINE
OB90 C5 1715 PUSH B
OB91 E9 1716 PCHL
OB92 14 46 19 1717 ACAL4 LXI D,CALST
    
```

OB95	CD	FA	0D	1718	CALL	CNS	
OB98	3E	0D		1719	MVI	A,CR	
OB9A	12			1720	STAX	D	
OB9B	11	46	19	1721	LXI	D,CALST	
OB9E	21	BB	18	1722	LXI	H,FPSIN	
OBA1	CD	57	12	1723	CALL	FPIN	
OBA4	11	BB	18	1724	LXI	D,FPSIN	
OBA7	C3	A8	0C	1725	JMP	PSHA1	PUT THE RETURNED USER VALUE ON ARG STACK
OBAA				1726	*		
OBAA				1727	*	INT FUNCTION ACTION ROUTINE	
OBAA				1728	*		
OBAA	0A			1729	AINT	LDAX	B
OBAB	D6	81		1730		SUI	429
OBAD	F2	B3	0B	1731		JP	AINT1
OBBO				1732	*	ZERO IF VALUE LESS THAN ONE	
OBBO	AF			1733		XRA	A
OBBI	02			1734		STAX	B
OBBI	C9			1735		RET	
OBBI				1736	*	EXP > 0	
OBBI	D6	05		1737	AINT1	SUI	FPNIB-1
OBBI	DO			1738		RNC	
OBBI	57			1739		MOV	D,A
OBBI	0B			1740		DCX	B
OBBI	0B			1741	AINT2	DCX	B
OBBI	0A			1742		LDAX	B
OBBI	E6	F0		1743		ANI	360Q
OBBI	02			1744		STAX	B
OBBI	14			1745		INR	D
OBBI	C8			1746		RZ	
OBBI	AF			1747		XRA	A
OBBI	02			1748		STAX	B
OBBI	14			1749		INR	D
OBBI	C2	B8	0B	1750		JNZ	AINT2
OBBI	C9			1751		RET	
OBBI				1752	*		
OBBI				1753	*	DIMENSION MATRIX	
OBBI				1754	*	SYMTAB ADDRESS IN HL, HL NOT CLOBBERED	
OBBI				1755	*	DE CONTAINS SIZE IN NUMBER OF ELEMENTS	
OBBI				1756	*		
OBBI	E5			1757	DIMS	PUSH	H
OBBI	13			1758		INX	D
OBBI	D5			1759		PUSH	D
OBBI	21	00	00	1760		LXI	H,0
OBBI	0E	05		1761		MVI	C,FPSIZ
OBBI	CD	18	0D	1762		CALL	RADD
OBBI	EB			1763		XCHG	
OBBI	2A	1E	19	1764		LHLD	MATA
OBBI	E5			1765		PUSH	H
OBBI	19			1766		DAD	D
OBBI	CD	E2	0C	1767		CALL	STOV
OBBI	22	1E	19	1768		SHLD	MATA
OBBI	C1			1769		POP	B
OBBI	D1			1770		POP	D
OBBI	E1			1771		POP	H
OBBI	E5			1772		PUSH	H
OBBI	72			1773		MOV	M,D

OBE2 2B	1774	DCX	H	
OBE3 73	1775	MOV	M,E	
OBE4 2B	1776	DCX	H	
OBE5 70	1777	MOV	M,B	
OBE6 2B	1778	DCX	H	
OBE7 74	1779	MOV	M,C	SYMTAB ENTRY NOW SET UP
OBE8 E1	1780	POP	H	
OBE9 C9	1781	RET		
OBEA	1782 *			
OBEA	1783 *	FIND VARIABLE OPTIONALLY SUBSCRIBED IN TEXT		
OBEA	1784 *	SETS CARRY IF NOT FOUND		
OBEA	1785 *	RETURNS ADDRESS OF VARIABLE IN HL		
OBEA	1786 *	UPDATES TXA IF FOUND		
OBEA	1787 *			
OBEA CD 80 OC	1788	VAR	CALL	ALPHA
OBED D8	1789		RC	
OBEE CD 4D OC	1790		CALL	NAME2
OBF1 CD 02 OD	1791		CALL	GC
OBF4 FE E0	1792		CPI	LPARRW
OBF6 CA FE OB	1793		JZ	VAR1
OBF9	1794 *	MUST BE SCALAR VARIABLE		
OBF9 CD 58 OC	1795		CALL	STLK
OBFC B7	1796		ORA	A
OBFD C9	1797		RET	
OBFE	1798 *	MUST BE SUBSCRIBED		
OBFE CD 0A OD	1799	VAR1	CALL	GCI
OC01 3E 80	1800		MVI	A,200Q
OC03 B1	1801		ORA	C
OC04 4F	1802		MOV	C,A
OC05 CD 58 OC	1803		CALL	STLK
OC08 E5	1804		PUSH	H
OC09 14 0A 00	1805		LXI	D,40
OC0C DC C6 OB	1806		CC	DIMS
OC0F CD D7 05	1807		CALL	EXPRB
OC12 CD 8A OD	1808		CALL	PFIX
OC15 06 29	1809		MVI	B,')'
OC17 CD FA OC	1810		CALL	EATC
OC1A E1	1811		POP	H
OC1B 2B	1812		DCX	H
OC1C CD 35 OD	1813		CALL	DCMP
OC1F D2 BC 00	1814		JNC	E5
OC22 2B	1815		DCX	H
OC23 2B	1816		DCX	H
OC24 CD 40 OD	1817		CALL	LHLI
OC27 0E 05	1818		MVI	C,FPSIZ
OC29 13	1819		INX	D
OC2A CD 48 OD	1820		CALL	RADD
OC2D C9	1821		RET	
OC2E	1822 *			
OC2E	1823 *	JUNK ON END OF STATEMENT, TEST IF AT END OF FILE		
OC2E	1824 *	DOES NOT CLOBBER DE		
OC2E	1825 *	EATS CHARACTER AND LINE COUNT AFTER CR		
OC2E	1826 *	LEAVES NEW TXA IN HL		
OC2E	1827 *	SETS CARRY IF END OF FILE		
OC2E	1828 *			
OC2E CD 0A OD	1829	JOE	CALL	GCI

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0C31	FE 3B	1830	CPI	','	
0C33	C8	1831	RZ		
0C34	FE 0D	1832	CPI	CR	
0C36	C2 AA 00	1833	JNZ	E4	
0C39	7E	1834	MOV	A,M	
0C3A	3D	1835	DCR	A	
0C3B	CA 45 0C	1836	JZ	JOE2	
0C3E	23	1837	INX	H	
0C3F	23	1838	INX	H	
0C40	23	1839	INX	H	SKIP OVER COUNT AND LINE NUMBER
0C41	22 95 17	1840	JOE1 SHLD	TXA	
0C44	C9	1841	RET		
0C45	37	1842	JOE2	STC	
0C46	C3 41 0C	1843	JMP	JOE1	
0C49		1844	*		
0C49		1845	*	GET NAME FROM TEXT	
0C49		1846	*	SETS CARRY IF NAME NOT FOUND	
0C49		1847	*	IF SUCCEEDS RETURNS NAME IN BC, C=0 IF NO DIGIT IN NAME	
0C49	CD 80 0C	1848	NAME CALL	ALPHA	
0C4C	D8	1849	RC		
0C4D	47	1850	NAME2 MOV	B,A	
0C4E	0E 00	1851	MVI	C,0	
0C50	CD 8D 0C	1852	CALL	DIG	
0C53	3F	1853	CMC		
0C54	D0	1854	RNC		
0C55	4F	1855	MOV	C,A	
0C56	B7	1856	ORA	A	CLEAR CARRY
0C57	C9	1857	RET		
0C58		1858	*		
0C58		1859	*	SYMBOL TABLE LOOKUP	
0C58		1860	*	BC CONTAIN NAME AND CLASS	
0C58		1861	*	IF NOT FOUND THEN CREATE ZERO'ED ENTRY AND SET CARRY	
0C58		1862	*	HL HAS ADDRESS ON RET	
0C58		1863	*		
0C58	2A 52 19	1864	STLK LHLD	MEMTOP	
0C5B	11 F9 FF	1865	LXI	D,-STESIZ	SET UP BASE AND INCREMENT FOR SEARCH LOOP
0C5E	7E	1866	STLK0 MOV	A,M	
0C5F	B7	1867	ORA	A	
0C60	CA 72 0C	1868	JZ	STLK2	TEST IF END OF TABLE
0C63	B8	1869	CMP	B	
0C64	C2 6E 0C	1870	JNZ	STLK4	TEST IF ALPHA COMPARES
0C67	2B	1871	DCX	H	
0C68	7E	1872	MOV	A,M	LOOK FOR DIGIT
0C69	B9	1873	CMP	C	
0C6A	2B	1874	DCX	H	
0C6B	C8	1875	RZ	.	CARRY CLEAR OO RET
0C6C	23	1876	INX	H	
0C6D	23	1877	INX	H	
0C6E	19	1878	STLK1 DAD	D	DIDN'T COMPARE, DECREMENT POINTER
0C6F	C3 5E 0C	1879	JMP	STLK0	
0C72		1880	*	ADD ENTRY TO SYMTAB	
0C72	70	1881	STLK2 MOV	M,B	
0C73	2B	1882	DCX	H	
0C74	71	1883	MOV	M,C	
0C75	23	1884	INX	H	
0C76	EB	1885	XCHG		

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0C77 19	1886	DAD	D	
0C78 22 B3 18	1887	SHLD	STA	STORE NEW END OF SIMTAB POINTER
0C7B 1B	1888	DCX	D	
0C7C 1B	1889	DCX	D	
0C7D EB	1890	XCHG		
0C7E 37	1891	STC		
0C7F C9	1892	RET		
0C80	1893	*		
0C80	1894	*		GOBBLES NEXT TEXT CHARACTER IF ALPHABETIC
0C80	1895	*		SETS CARRY IF NOT
0C80	1896	*		NEXT CHAR IN ACC ON FAILURE
0C80	1897	*		
0C80 CD 02 OD	1898	ALPHA CALL	GC	
0C83 FE 41	1899	CPI	'A'	
0C85 D8	1900	RC		
0C86 FE 5B	1901	CPI	'Z'+1	
0C88 3F	1902	CMC		
0C89 D8	1903	RC		
0C8A C3 97 OC	1904	JMP	DIGT1	
0C8D	1905	*		GOBBLES NEXT TEXT CHAR IF DIGIT
0C8D	1906	*		SETS CARRY IF NOT
0C8D	1907	*		NEXT CHAR IN ACC ON FAILURE
0C8D CD 02 OD	1908	DIG CALL	GC	
0C90 FE 30	1909	CPI	'0'	
0C92 D8	1910	RC		
0C93 FE 3A	1911	CPI	'9'+1	
0C95 3F	1912	CMC		
0C96 D8	1913	RC		
0C97 23	1914	DIGT1 INX	H	
0C98 22 95 17	1915	SHLD	TXA	
0C9B C9	1916	RET		
0C9C	1917	*		
0C9C	1918	*		COPYS FPSIZ BYTES AT ADDR HL TO ADDR DE
0C9C	1919	*		ON EXIT HL POINTS TO ADR-1 OF LAST BYTE COPIED
0C9C	1920	*		
0C9C 0E 05	1921	VCOPY MVI	C, FPSIZ	
0C9E 7E	1922	VCOPY MOV	A, M	
0C9F 12	1923	STAX	D	
OCA0 2B	1924	DCX	H	
OCA1 1B	1925	DCX	D	
OCA2 0D	1926	DCR	C	
OCA3 C2 9E OC	1927	JNZ	VCOPY	
OCA6 C9	1928	RET		
OCA7	1929	*		
OCA7	1930	*		PUSH VALUE ADDRESSED BY HL ONTO ARG STACK
OCA7	1931	*		SETS ARGF, CLEARS CARRY
OCA7	1932	*		
OCA7 EB	1933	PSHAS XCHG		
OCA8 2A 1C 19	1934	PSHA1 LHLD	ASTKA	
OCA8 01 FB FF	1935	LXI	B, -FPSIZ	
OCAE 09	1936	DAD	B	
OCAF 22 1C 19	1937	SHLD	ASTKA	DECREMENT ARG STACK POINTER
OCB2 EB	1938	XCHG		
OCB3 CD 9C OC	1939	CALL	VCOPY	
OCB6 3E 01	1940	MVI	A, 1	
OCB8 32 93 17	1941	STA	ARGF	CLEAR ARGF

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

OCBB B7	1942	ORA	A	CLEAR CARRY
OCBC C9	1943	RET		
OCBD	1944 *			
OCBD	1945 *	POP ARG STACK		
OCBD	1946 *	HL CONTAINS ADDRESS TO PUT POPED VALUE AT		
OCBD	1947 *			
OCBD EB	1948	POPAS	XCHG	
OCBE 2A 4C 49	1949	POPAN	LHLD	ASTKA
OCC1 E5	1950		PUSH	H
OCC2 04 05 00	1951		LXI	B,FPSIZ
OCC5 09	1952		DAD	B
OCC6 22 4C 49	1953		SHLD	ASTKA INCREMENT STACK POINTER
OCC9 E4	1954		POP	H
OCCA C3 9C 0C	1955		JMP	VCOPY
OCCD	1956 *			
OCCD	1957 *	PUSH FRAME ONTO CONTROL STACK		
OCCD	1958 *	TAKES MINUS AMOUNT TO SUB FROM CSTKA IN DE		
OCCD	1959 *	DOES OVERFLOW TEST AND RETURNS OLD CSTKA-1		
OCCD	1960 *			
OCCD 2A B5 48	1961	PSHCS	LHLD	CSTKA
OCDO E5	1962		PUSH	H
OCDA 49	1960		DAD	D
OCDA 22 B5 48	1961		SHLD	CSTKA
OCDA 5B	1962		XCHG	
OCDA 24 D0 07	1963		LXI	H,LCSTKA ADDR CONTAINS CSTKL
OCDA 9D 35 0D	1964		CALL	DCMP
OCDC DA B6 00	1965		JC	E4
OCDF E4	1966		POP	H
OCE0 2B	1967		DCX	H
OCE1 C9	1968		RET	
OCE2	1969 *			
OCE2	1970 *	STORAGE OVERFLOW TEST		
OCE2	1971 *	TEST THAT VALUE IN HL IS BETWEEN MATA AND STA		
OCE2	1972 *	DOES NOT CLOBBER HL		
OCE2	1973 *			
OCE2 EB	1974	STOV	XCHG	
OCE3 24 1E 49	1975		LXI	H,MATA
OCE6 CD 35 0D	1976		CALL	DCMP
OCE9 DA F4 0C	1977		JC	E8
OCEC 24 B3 48	1978		LXI	H,STA
OCEF CD 35 0D	1979		CALL	DCMP
OCF2 EB	1980		XCHG	
OCF3 D8	1981		RC	
OCF4 01 4F 53	1982	E8	LXI	B,'SO'
OCF7 C3 C5 00	1983		JMP	ERROR
OCFA	1984 *			
OCFA	1985 *	INCREMENT TXA IF NEXT NON-BLANK CHAR IS EQUAL TO B		
OCFA	1986 *	ELSE SYNTAX ERROR		
OCFA	1987 *			
OCFA CD 0A 0D	1988	EATC	CALL	GCI
OCFD B8	1989		CMP	B
OCFE C8	1990		RZ	
OCFF C3 AA 00	1991		JMP	E4
OD02	1992 *			
OD02	1993 *	GET NEXT NON-BLANK CHAR INTO ACC		
OD02	1994 *	INCREMENT PAST BLANKS ONLY		

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0D02		1995 *			
0D02	CD OA OD	1996 GC	CALL	GCI	
0D05	2B	1997	DCX	H	
0D06	22 95 17	1998	SHLD	TXA	
0D09	C9	1999	RET		
0D0A		2000 *			
0D0A		2001 *	GET NEXT NON-BLANK TEXT CHAR AND INCREMENT TXA		
0D0A		2002 *	DOES NOT CLOBBER DE, BC		
0D0A		2003 *	RETURN CHAR IN ACC		
0D0A		2004 *			
0D0A	2A 95 17	2005 GCI	LHLD	TXA	
0D0D	7E	2006 GCIO	MOV	A,M	
0D0E	23	2007	INX	H	
0D0F	FE 20	2008	CPI	' '	
0D11	CA OD OD	2009	JZ	GCIO	
0D14	22 95 17	2010	SHLD	TXA	
0D17	C9	2011	RET		
0D18		2012 *			
0D18		2013 *	REPEAT ADD		
0D18		2014 *	ADDS DE TO HL C TIMES		
0D18		2015 *			
0D18	19	2016 RADD	DAD	D	
0D19	OD	2017	DCR	C	
0D1A	C2 18 OD	2018	JNZ	RADD	
0D1D	C9	2019	RET		
0D1E		2020 *			
0D1E		2021 *	PRINT MESSAGE ADDRESSED BY HL		
0D1E		2022 *	ENDS WITH CHARACTER PROVIDED IN C		
0D1E		2023 *	RETURN IN HL ADDRESS OF TERMINATOR		
0D1E		2024 *			
0D1E	OE OD	2025 PRN1	MVI	C,CR	
0D20	C3 25 OD	2026	JMP	PRN1	
0D23	OE 22	2027 PRN1	MVI	C,'''	
0D25	7E	2028 PRN1	MOV	A,M	GET NEXT CHAR
0D26	47	2029	MOV	B,A	FOR CHOUT
0D27	B9	2030	CMP	C	END OF MESSAGE TEST
0D28	C8	2031	RZ		
0D29	FE OD	2032	CPI	CR	
0D2B	CA AA OD	2033	JZ	E1	NEVER PRINT A CR IN THIS ROUTINE
0D2E	CD 95 OE	2034	CALL	CHOUT	
0D31	23	2035	INX	H	
0D32	C3 25 OD	2036	JMP	PRN1	
0D35		2037 *			
0D35		2038 *	16 BIT UNSIGNED COMPARE		
0D35		2039 *	COMPARE DE AGAINST VALUE ADDRESSED BY HL		
0D35		2040 *	CLOBBERS A ONLY		
0D35		2041 *			
0D35	7B	2042 DCMP	MOV	A,E	
0D36	96	2043	SUB	M	
0D37	23	2044	INX	H	
0D38	7A	2045	MOV	A,D	
0D39	9E	2046	SBB	M	
0D3A	2B	2047	DCX	H	
0D3B	C0	2048	RNZ		
0D3C	7B	2049	MOV	A,E	
0D3D	96	2050	SUB	M	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

0D3E B7	2051	ORA	A	CLEAR CARRY
0D3F C9	2052	RET		
0D40	2053 *			
0D40	2054 *	INDIRECT LOAD HL THRU HL		
0D40	2055 *			
0D40 F5	2056	LHLI	PUSH	PSW
0D41 7E	2057	MOV	A,M	
0D42 23	2058	INX	H	
0D43 66	2059	MOV	H,M	
0D44 6F	2060	MOV	L,A	
0D45 F4	2061	POP	PSW	
0D46 C9	2062	RET		
0D47	2063 *			
0D47	2064 *	GET FP CONSTANT FROM TEXT		
0D47	2065 *	PUSHES VALUE ON ARG STACK AND SETS ARGF FLAG		
0D47	2066 *	SETS CARRY IF NOT FOUND		
0D47	2067 *			
0D47 2A 95 17	2068	CONST	LHLD	TXA PREPARE CALL FPIN
0D4A EB	2069		XCHG	
0D4B 24 BB 18	2070		LXI	H,FPSINK
0D4E CD 57 12	2071		CALL	FPIN
0D51 D8	2072		RC	
0D52 1B	2073		DCX	D
0D53 EB	2074		XCHG	
0D54 22 95 17	2075		SHLD	TXA NOW POINTS TO TERMINATOR
0D57 11 BB 18	2076		LXI	D,FPSINK
0D5A CD A8 0C	2077		CALL	PSHA1
0D5D AF	2078		XRA	A
0D5E 3C	2079		INR	A SET A TO 1 AND CLEAR CARRY
0D5F 32 93 17	2080		STA	ARGF
0D62 C9	2081		RET	
0D63	2082 *			
0D63	2083 *	DIRECT STATEMENT CHECKING ROUTINE		
0D63	2084 *			
0D63 3A 94 17	2085	DIRT	LDA	DIRF
0D66 B7	2086		ORA	A
0D67 C8	2087		RZ	
0D68 01 49 44	2088		LXI	B,'DI'
0D6B C3 C5 00	2089		JMP	ERROR
0D6E	2090 *			
0D6E	2091 *	FIND TEXT LINE WITH LINE NUMBER GIVEN IN DE		
0D6E	2092 *	RETURNS TEXT ADDRESS COUNT BYTE IN HL		
0D6E	2093 *			
0D6E 2A 50 19	2094	FINDLN	LHLD	BOFA
0D71 06 00	2095		MVI	B,0
0D73 4E	2096	FIND1	MOV	C,M
0D74 79	2097		MOV	A,C
0D75 FE 01	2098		CPI	EOF
0D77 CA 84 0D	2099		JZ	LERR
0D7A 23	2100		INX	H
0D7B CD 35 0D	2101		CALL	DCMP
0D7E 2B	2102		DCX	H
0D7F C8	2103		RZ	
0D80 09	2104		DAD	B
0D81 C3 73 0D	2105		JMP	FIND1
0D84 01 4E 4C	2106	LERR	LXI	B,'LN'

0D87 C3 C5 00	2107	JMP	ERROR	
0D8A	2108 *			
0D8A	2109 *	FIX FLOATING TO POSITIVE INTEGER		
0D8A	2110 *	RETURN INTEGER VALUE IN DE		
0D8A	2111 *	FP VALUE FROM TOP OF ARG STACK, POP ARG STACK		
0D8A	2112 *			
0D8A 2A 1C 19	2113	PFIX	LHLD	ASTKA
0D8D 44	2114		MOV	B,H
0D8E 4D	2115		MOV	C,L
0D8F E5	2116		PUSH	H
0D90 CD AA 0B	2117		CALL	AINT
0D93 21 BB 18	2118		LXI	H,FPSPINK
0D96 CD BD 0C	2119		CALL	POPAS
0D99 E1	2120		POP	H
0D9A 4E	2121		MOV	C,M EXPONENT
0D9B 2B	2122		DCX	H
0D9C 7E	2123		MOV	A,M SIGN
0D9D B7	2124		ORA	A
0D9E C2 BC 00	2125		JNZ	E5 NEGATIVE NO GOOD
0DA1 11 FC FF	2126		LXI	D,-FPSIZ+1
0DA4 19	2127		DAD	D
0DA5 11 00 00	2128		LXI	D,0
0DA8 79	2129		MOV	A,C
0DA9 B7	2130		ORA	A
0DAA C8	2131		RZ	
0DAB 0D	2132		DCR	C SET UP FOR LOOP CLOSE TEST
0DAC 23	2133	PFIX1	INX	H
0DAD 7E	2134		MOV	A,M
0DAE 0F	2135		RRC	
0DAF 0F	2136		RRC	
0DB0 0F	2137		RRC	
0DB1 0F	2138		RRC	
0DB2 CD C6 0D	2139		CALL	MUL10
0DB5 DA BC 00	2140		JC	E5
0DB8 0D	2141		DCR	C
0DB9 F0	2142		RP	
0DBA 7E	2143		MOV	A,M
0DBB CD C6 0D	2144		CALL	MUL10
0DBE DA BC 00	2145		JC	E5
0DC1 0D	2146		DCR	C
0DC2 FA AC 0D	2147		JM	PFIX1
0DC5 C9	2148		RET	
0DC6	2149 *			
0DC6	2150 *	TAKE NEXT DIGIT IN A (MASK TO 17Q), ACCUMULATE TO DE		
0DC6	2151 *	PRESERVES ALL BUT A, DE		
0DC6	2152 *			
0DC6 E5	2153	MUL10	PUSH	H
0DC7 33	2154		INX	SP
0DC8 33	2155		INX	SP
0DC9 62	2156		MOV	H,D GET ORIGINAL VALUE TO HL
0DCA 6B	2157		MOV	L,E
0DCB 29	2158		DAD	H DOUBLE IT
0DCC D8	2159		RC	
0DCD 29	2160		DAD	H ACAIN
0DCE D8	2161		RC	
0DCF 19	2162		DAD	D PLUS ORIGINAL MAKES 5 TIMES ORIG

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

ODD0 D8	2163	RC		
ODD1 29	2164	DAD	H	TIMES TWO MAKES TEN
ODD2 D8	2165	RC		
ODD3 EB	2166	XCHG		
ODD4 3B	2167	DCX	SP	
ODD5 3B	2168	DCX	SP	
ODD6 E1	2169	POP	H	
ODD7 E6 OF	2170	ANI	17Q	
ODD9 83	2171	ADD	E	
ODDA 5F	2172	MOV	E,A	
ODDB 7A	2173	MOV	A,D	
ODDC CE 00	2174	ACI	0	PROPOGATE THE CARRY
ODDE 57	2175	MOV	D,A	
ODDF C9	2176	RET		
ODE0	2177	*		
ODE0	2178	*	GET INTEGER FROM TEXT	
ODE0	2179	*	SET CARRY IF NOT FOUND	
ODE0	2180	*	RETURN INTEGER VALUE IN HL	
ODE0	2181	*	RETURN TERMINATOR IN ACC	
ODE0	2182	*		
ODE0 CD 8D 0C	2183	INTGER	CALL	DIG
ODE3 D8	2184	RC		
ODE4 14 00 00	2185	LXI	D,0	
ODE7 C3 F3 0D	2186	JMP	INTG2	
ODEA CD 8D 0C	2187	INTG1	CALL	DIG
ODED 62	2188	MOV	H,D	
ODEE 6B	2189	MOV	L,E	
ODEF 3F	2190	CMC		
ODF0 D0	2191	RNC		
ODF4 D6 30	2192	INTG2	SUI	'0'
ODF3 CD C6 0D	2193	CALL	MUL10	
ODF6 D2 EA 0D	2194	JNC	INTG1	
ODF9 C9	2195	RET		
ODFA	2196	*		
ODFA	2197	*	CONVERT INTEGER TO STRING	
ODFA	2198	*	DE CONTAINS ADDRESS OF STRING, RETURN UPDATED VALUE IN DE	
ODFA	2199	*	HL CONTAINS VALUE TO CONVERT	
ODFA	2200	*		
ODFA AF	2201	CNS	XRA	A SET FOR NO LEADING ZEROES
ODFB 01 F0 D8	2202	LXI	B,-10000	
ODFE CD 4F 0E	2203	CALL	RSUB	
OE01 01 48 FC	2204	LXI	B,-1000	
OE04 CD 4F 0E	2205	CALL	RSUB	
OE07 01 9C FF	2206	LXI	B,-100	
OE0A CD 1F 0E	2207	CALL	RSUB	
OE0D 01 F6 FF	2208	LXI	B,-10	
OE10 CD 4F 0E	2209	CALL	RSUB	
OE13 01 FF FF	2210	LXI	B,-1	
OE16 CD 4F 0E	2211	CALL	RSUB	
OE19 C0	2212	RNZ		
OE1A 3E 30	2213	MVI	A,'0'	
OE1C 42	2214	STAX	D	
OE1D 13	2215	INX	D	
OE1E C9	2216	RET		
OE1F	2217	*		
OE1F	2218	*	TAKE VALUE IN HL	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

OE4F	2219	*	SUB MINUS NUMBER IN BE THE MOST POSSIBLE TIMES	
OE4F	2220	*	PUT VALUE ON STRING AT DE	
OE4F	2221	*	IF A=0 THEN DONT PUT ZERO ON STRING	
OE4F	2222	*	RETURN NON-ZERO IN A IF PUT ON STRING	
OE4F	2223	*		
OE4F D5	2224	RSUB	PUSH	D
OE20 16 FF	2225		MVI	D,-1
OE22 E5	2226	RSUB4	PUSH	H
OE23 33	2227		INX	SP
OE24 33	2228		INX	SP
OE25 14	2229		INR	D
OE26 09	2230		DAD	B
OE27 DA 22 OE	2231		JC	RSUB4
OE2A 3B	2232		DCX	SP
OE2B 3B	2233		DCX	SP
OE2C E4	2234		POP	H
OE2D 42	2235		MOV	B,D
OE2E D1	2236		POP	D
OE2F B0	2237		ORA	B
OE30 C8	2238		RZ	A GETS 0 IF A WAS 0 AND B IS 0
OE31 3E 30	2239		MVI	A,'Q'
OE33 80	2240		ADD	B
OE34 12	2241		STAX	D
OE35 13	2242		INX	D
OE36 C9	2243		RET	
OE37	2244	*		
OE37	2245	*	INPUT CHARACTER FROM TERMINAL	
OE37	2246	*		
OE37 CD 90 OE	2247	INCHAR	CALL	STATUS
OE3A CA 37 OE	2248		JZ	INCHAR
OE3D DB 04	2249		IN	LCOP UNTIL CHAR RECEIVED
OE3F E6 7F	2250		ANI	477Q
OE41 FE 03	2251		CPI	ESC
OE43 CA 62 00	2252		JZ	CMND4
OE46 FE 0A	2253		CPI	LF
OE48 CA 37 OE	2254		JZ	INCHAR
OE4B FE 00	2255		CPI	0
OE4D CA 37 OE	2256		JZ	INCHAR
OE50 47	2257		MOV	B,A
OE51 C9	2258		RET	
OE52	2259	*		
OE52 CD 4C 40	2260	INLO	CALL	CRLF
OE55 24 D3 18	2261	INLINE	LXI	H,IBUF
OE58 OE 49	2262		MVI	C,LINLEN
OE5A CD 37 OE	2263	INL4	CALL	INCHAR
OE5D FE 5F	2264		CPI	5FH
OE5F CA 7E OE	2265		JZ	INL2
OE62 77	2266		MOV	M,A
OE63 CD 95 OE	2267		CALL	CHOUT
OE66 78	2268		MOV	A,B
OE67 FE 40	2269		CPI	'@'
OE69 CA 52 OE	2270		JZ	INLO
OE6C 06 0A	2271		MVI	B,LF
OE6E FE 0D	2272		CPI	CR
OE70 CA 95 OE	2273		JZ	CHOUT
OE73 23	2274		INX	H

0E74	0D	2275	DCR	C	
0E75	C2 5A 0E	2276	JNZ	INL4	
0E78	03 4C 4C	2277	LXI	B,'LL'	
0E7B	C3 C5 00	2278	JMP	ERROR	
0E7E	79	2279	INL2	MOV	A,C
0E7F	06 07	2280		MVI	B,BELL
0E81	FE 49	2281		CPI	LINLEN
0E83	CA 8A 0E	2282		JZ	INL3
0E86	06 5F	2283		MVI	B,5FH
0E88	2B	2284		DCX	H
0E89	0C	2285		INR	C
0E8A	CD 95 0E	2286	INL3	CALL	CHOUT
0E8D	C3 5A 0E	2287		JMP	INL4
0E90		2288	*		
0E90	DB 00	2289	STATUS	IN	0
0E92	E6 40	2290		ANI	40H
0E94	C9	2291		RET	
0E95		2292	*		
0E95		2293	*		OUTPUT ROUTINES
0E95		2294	*		SENSE SWITCH 4 (A8) CONTROLS VDM OR STANDARD TERMINAL
0E95		2295	*		
0E95	DB FF	2296	CHOUT	IN	OFFH GET SENSE SWITCH INPUT
0E97	1F	2297		RAR	
0E98	DA E4 0F	2298		JC	TEROT SWITCH UP IS STANDARD TERMINAL
0E9B		2299	*		
0E9B		2300	*		
0E9B		2301	*		OUTPUT DRIVER FOR PROCESSOR TECHNOLOGY VIDEO DISPLAY
0E9B		2302	*		MODULE. SEE VDM MANUAL FOR ADDITIONAL CODE COMMENTS
0E9B		2303	*		
0E9B	78	2304	ALSO	MOV	A,B
0E9C	FE 7F	2305		CPI	7FH
0E9E	C8	2306		RZ	
0E9F	FE 07	2307		CPI	BELL
0EA1	CA E4 0F	2308		JZ	TEROT "DING" GOES THE STANDARD TERMINAL
0EA4		2309	*		
0EA4		2310	*		THIS ROUTINE ALLOWS AN OUTPUT RATE SELECTABLE FROM
0EA4		2311	*		THE KEYBOARD BETWEEN THREE CHARACTERS PER SECOND
0EA4		2312	*		TO APROX 2000 LINES PER MINUTE.
0EA4		2313	*		
0EA4	E5	2314	TIMER	PUSH	H
0EA5	D5	2315		PUSH	D
0EA6	C5	2316		PUSH	B
0EA7	3A 45 49	2317		LDA	SPEED GET DELAY TIME
0EAA	67	2318		MOV	H,A
0EAB	2E 80	2319		MVI	L,80H
0EAD	CD C3 0E	2320		CALL	NUMCK SEE IF NEW SPEED IS WANTED
0EB0	AF	2321		XRA	A
0EB1	2B	2322	TIME	DCX	H
0EB2	BC	2323		CMP	H ZERO YET?
0EB3	C2 B4 0E	2324		JNZ	TIME
0EB6	C1	2325		POP	B GET BACK CHR
0EB7	C5	2326		PUSH	B
0EB8	78	2327		MOV	A,B
0EB9	CD 03 0F	2328		CALL	SCOUT OUTPUT CHR TO SCREEN
0EBC	C1	2329		POP	B
0EBD	D1	2330		POP	D

OEBE E4	2334	POP	H	
OEBF 78	2332	MOV	A,B	
OECO C3 EE OF	2333	JMP	CHCHK	NOW PROCESS THE REST
OEC3	2334	*		
OEC3	2335	*	CHECK FOR TIMER CONTROL VALUE	
OEC3	2336	*		
OEC3 3A 3F 19	2337	NUMCK LDA	BRKCHR	
OEC6 47	2338	MOV	B,A	
OEC7 B7	2339	ORA	A	
OEC8 CC 90 OE	2340	CZ	STATUS	
OECB CA D3 OE	2341	JZ	NINP	
OECE DB 01	2342	IN	1	
OEDO E6 7F	2343	ANI	7FH	
OED2 47	2344	MOV	B,A	
OED3 78	2345	NINP MOV	A,B	
OED4 B7	2346	NUMC2 ORA	A	
OED5 C8	2347	.		NO NEW INPUT VALUE
OED6 FE 3A	2348	CPI	'9'+1	
OED8 D2 F2 OE	2349	JNC	WAIT	
OEDB FE 34	2350	CPI	'1'	
OEDD DA F2 OE	2351	JC	WAIT	
OEEO E6 OF	2352	ANI	OFH	REMOVE ASCII BIAS
OEED 4F	2353	MOV	C,A	SAVE DELAY NUMBER
OEED 37	2354	XRA	A	
OEED 32 45 19	2355	STC	.	INITIALIZE DELAY BIT IN CARRY
OEED 17	2356	LESS STA	SPEED	
OEED 0D	2357	RAL		
OEED C2 E5 OE	2358	DCR	C	
OEED AF	2359	JNZ	LESS	
OEED 32 3F 19	2360	CBRK XRA	A	
OEED C9	2361	SBRK STA	BRKCHR	
OEED 20	2362	RET		
OEED FE 20	2363	*		
OEED DA EE OE	2364	WAIT CPI	20H	
OEED C2 ED OE	2365	JC	SBRK	
OEED CD 90 OE	2366	JNZ	CBRK	
OEED CA FA OE	2367	WAIT2 CALL	STATUS	WAIT FOR KEYBOARD INPUT
OEED C3 EE OE	2368	JZ	WAIT2	
OEED 4F	2369	JMP	SBRK	
OEED 4F	2370	*		
OEED 4F	2371	*	OUTPUT CHR IN REG A TO SCREEN	
OEED 4F	2372	*		
OEED 21 41 19	2373	SCOUT MOV	C,A	
OEED 46	2374	LXI	H,CCP	GET CURRENT CURSOR POSITION
OEED FE OD	2375	MOV	B,M	
OEED CA 2F OF	2376	CPI	CR	CARRIAGE RETURN?
OEED FE 5F	2377	JZ	SCOT2	
OEED CA 58 OF	2378	CPI	5FH	BACKSPACE?
OEED FE 01	2379	JZ	BKSPA	
OEED CA 8E OF	2380	CPI	'A'-40H	CURSOR ON-OFF
OEED FE 1A	2381	JZ	CURTG	
OEED CA 68 OF	2382	CPI	'Z'-40H	CLEAR SCREEN
OEED FE 20	2383	JZ	INITS	
OEED D8	2384	CPI	20H	
OEED 4F	2385	RC	.	NO OTHER CONTROL CHRS
OEED 4F	2386	*		

OF1F	2387	*	STORE CHR IN VDM MEMORY	
OF1F	2388	*		
OF1F 3A 40 19	2389		LDA CLN	CURRENT LINE NUMBER
OF22 CD C9 OF	2390		CALL CLNA	FIND ADDRESS
OF25 74	2391		MOV M,C	PUT IN THE CHR
OF26 3A 44 19	2392		LDA CCP	ADVANCE THE CURSOR
OF29 3C	2393		INR A	
OF2A FE 40	2394		CPI 64	LAST CHR IN SCREEN LINE?
OF2C C2 39 OF	2395		JNZ SCOT1	
OF2F	2396	*	ADVANCE CURSOR LINE	
OF2F 3A 40 19	2397	SCOT2	LDA CLN	
OF32 CD DC OF	2398		CALL CCUR	
OF35 CD A0 OF	2399		CALL SCRL	
OF38 97	2400		SUB A	
OF39 32 44 19	2401	SCOT4	STA CCP	
OF3C 47	2402		MOV B,A	
OF3D 3A 40 19	2403		LDA CLN	
OF40 E6 OF	2404	SCUR	ANI OFH	
OF42 32 40 19	2405		STA CLN	
OF45 CD C9 OF	2406		CALL CLNA	
OF48 78	2407		MOV A,B	
OF49 32 44 19	2408		STA CCP	
OF4C 3A 42 19	2409		LDA CURF	
OF4F B7	2410		ORA A	
OF50 7E	2411		MOV A,M	
OF51 CA E0 OF	2412		JZ CCUR2	
OF54 F6 80	2413		ORI 80H	
OF56 77	2414		MOV M,A	
OF57 C9	2415		RET	
OF58	2416	*	BACKSPACE AND ERASE LAST CHR	
OF58 3A 40 19	2417	BKSPA	LDA CLN	
OF5B CD DC OF	2418		CALL CCUR	
OF5E 2B	2419		DCX H	
OF5F 36 20	2420		MVI M,' '	BLANK IT
OF64 05	2421		DCR B	
OF62 3A 40 19	2422		LDA CLN	
OF65 C3 40 OF	2423		JMP SCUR	
OF68	2424	*		
OF68	2425	*	CLEAR SCREEN AND INITIALIZE PARAMETERS	
OF68	2426	*		
OF68 24 00 CC	2427	INITS	LXI H,VDMBASE	
OF6B 01 00 04	2428		LXI B,4024	
OF6E 36 20	2429	IL2	MVI M,' '	
OF70 23	2430		INX H	
OF71 0B	2431		DCX B	
OF72 78	2432		MOV A,B	
OF73 B4	2433		ORA C	
OF74 C2 6E OF	2434		JNZ IL2	LOOP UNTIL ALL 4K CHARACTER PLACES HAVE A SPACE
OF77	2435	*	SCREEN IS CLEAR NOW SET PARAMETERS	
OF77 32 43 19	2436		STA BOSL	
OF7A 32 44 19	2437		STA BOTL	
OF7D 32 90 17	2438		STA PHEAD	
OF80 32 44 19	2439		STA CCP	
OF83 32 42 19	2440		STA CURF	
OF86 3E OF	2441		MVI A,15	
OF88 32 40 19	2442		STA CLN	

PROCESSOR TECHNOLOGY BASIC 5
 *** COPYRIGHT 1976 ***

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

OF8B CD BB OF	2443	CALL	VDMOT	
OF8E	2444	* TOGGLE CURSOR	(ON--OFF---ON.....)	
OF8E 3A 42 19	2445	CURTG LDA	CURF	
OF91 EE 01	2446	XRI	1	
OF93 32 42 19	2447	STA	CURF	
OF96 3A 41 19	2448	LDA	CCP	
OF99 47	2449	MOV	B,A	
OF9A 3A 40 19	2450	LDA	CLN	
OF9D C3 40 OF	2451	JMP	SCUR	
OFA0	2452	*		
OFA0 21 44 19	2453	SCRL LXI	H,BOTL	
OFA3 E5	2454	PUSH	H	
OFA4 7E	2455	MOV	A,M	
OFA5 34	2456	INR	M	
OFA6 96	2457	SUB	M	
OFA7 01 00 00	2458	LXI	B,0	
OFAA CD C9 OF	2459	CALL	CLNA	
OFAD 01 40 20	2460	LXI	B,2040H	
OFB0 70	2461	SCRL2 MOV	M,B	
OFB1 2C	2462	INR	L	
OFB2 0D	2463	DCR	C	
OFB3 C2 B0 OF	2464	JNZ	SCRL2	
OFB6 E4	2465	POP	H	
OFB7 7E	2466	MOV	A,M	
OFB8 E6 OF	2467	ANI	OFH	
OFBA 77	2468	MOV	M,A	
OFBB	2469	*		
OFBB 3A 43 19	2470	VDMOT LDA	BOSL	
OFBE 07	2471	RLC		
OFBF 07	2472	RLC		
OFC0 07	2473	RLC		
OFC1 07	2474	RLC		
OFC2 21 44 19	2475	LXI	H,BOTL	
OFC5 B6	2476	ORA	M	
OFC6 D3 C8	2477	OUT	VDMDEV	
OFC8 C9	2478	RET		
OFC9	2479	*		
OFC9	2480	*	CALCULATE SCREEN ADDRESS AND RETURN IN HL	
OFC9	2481	*		
OFC9 6F	2482	CLNA MOV	L,A	
OFC9 3A 44 19	2483	LDA	BOTL	
OFCD 85	2484	ADD	L	
OFCE OF	2485	RRC		
OFCE OF	2486	RRC		
OFD0 6F	2487	MOV	L,A	
OFD1 E6 03	2488	ANI	3	LOW MOD MATH
OFD3 C6 CC	2489	ADI	VDMPAGE	
OFD5 67	2490	MOV	H,A	
OFD6 7D	2491	MOV	A,L	
OFD7 E6 C0	2492	ANI	COCH	
OFD9 80	2493	ADD	B	
OFDA 6F	2494	MOV	L,A	
OFDB C9	2495	RET		
OFDC	2496	*		
OFDC CD C9 OF	2497	CCUR CALL	CLNA	
OFDF 7E	2498	MOV	A,M	

```

OFE0 E6 7F      2499 CCUR2   ANI      7FH
OFE2 77         2500          MOV      M,A
OFE3 C9         2501          RET
OFE4           2502 *
OFE4           2503 *   STANDARD TERMINAL OUTPUT DRIVER
OFE4           2504 *
OFE4 DB 00      2505 TEROT   IN       0
OFE6 E6 80      2506          ANI      80H
OFE8 CA 95 0E   2507          JZ       CHOUT
OFE8 78         2508          MOV      A,B
OFEC D3 04      2509          OUT      1           TYPE IT
OFEE FE 0D      2510 CHCHK   CPI      CR
OFF0 C2 F7 0F   2511          JNZ      CHLF       NOT CR IS IT LF?
OFF3 AF         2512          XRA     A
OFF4 C3 03 10   2513          JMP     PSTOR       RETURN PHEAD TO ZERO
OFF7           2514 *
OFF7 FE 0A      2515 CHLF   CPI      LF
OFF9 CA 07 10   2516          JZ       NULCH      IF LINEFEED PROCESS THE NULLS
OFFC FE 20      2517          CPI      40Q       NO PHEAD INC IF CONTROL CHAR
OFFE D8         2518          RC
OFFF 3A 90 17   2519          LDA     PHEAD
1002 3C         2520          INR    A
1003 32 90 17   2521 PSTOR   STA     PHEAD
1006 C9         2522          RET
1007           2523 *
1007 3A 92 17   2524 NULCH   LDA     NULLCT     OUTPUT NULL CHARS
100A B7         2525          ORA     A
100B C8         2526          RZ
100C C5         2527          PUSH   B
100D 4F         2528          MOV     C,A
100E 06 00      2529          MVI    B,NULL
1010 CD 95 0E   2530 CH2    CALL   CHOUT       OUTPUT COUNT "C" NULLS
1013 0D         2531          DCR    C
1014 C2 10 10   2532          JNZ    CH2
1017 C1         2533          POP    B
1018 C9         2534          RET
1019 CD 1C 10   2535 CRLF2   CALL   CRLF
101C 06 0D      2536 CRLF   MVI    B,CR
101E CD 95 0E   2537          CALL   CHOUT
1024 06 0A      2538          MVI    B,LF
1023 C3 95 0E   2539          JMP     CHOUT
1026           2540 *
1026           2541 *   CHECK IF PANIC CHARACTER HAS BEEN HIT
1026           2542 *
1026 3A 3F 19   2543 PCHECK  LDA     BRKCHR
1029 B7         2544          ORA     A
102A CC 90 0E   2545          CZ     STATUS
102D C8         2546          RZ
102E DB 04      2547          IN     1           GET LAST CHR INPUT
1030 E6 7F      2548          ANI    7FH
1032 FE 03      2549          CPI    ESC
1034 CA 52 04   2550          JZ     STOP1
1037 32 3F 19   2551          STA    BRKCHR
103A C9         2552          RET
103B           2553 *
103B           2554 *   GET INTEGER FROM TERMINAL
  
```

103B		2555	*	DE CONTAINS STRING TO PRINT FIRST	
103B		2556	*	HL HAS 1 LESS THAN ACCEPTABLE LOWER BOUND	
103B		2557	*	THIS ROUTINE GOES TO START IF BAD NUMBER	
103B		2558	*	INTEGER VALUE RETURNED IN HL	
103B		2559	*		
103B	E5	2560	GINT	PUSH	H
103C	EB	2561		XCHG	
103D	3A 90 17	2562		LDA	PHEAD
1040	B7	2563		ORA	A
1041	C4 1C 10	2564		CNZ	CRLF
1044	CD 23 0D	2565		CALL	PRNT
1047	CD 55 0E	2566		CALL	INLINE
104A	21 D3 18	2567		LXI	H,IBUF
104D	22 95 17	2568		SHLD	TXA
1050	CD E0 0D	2569		CALL	INTGER
1053	DA 00 00	2570		JC	START
1056	FE 0D	2571		CPI	CR
1058	C2 00 00	2572		JNZ	START
105B	D1	2573		POP	D
105C	22 D3 18	2574		SHLD	IBUF USE IBUF AS A TEMP
105F	21 D3 18	2575		LXI	H,IBUF
1062	CD 35 0D	2576		CALL	DCMP
1065	D2 00 00	2577		JNC	START
1068	2A D3 18	2578		LHLD	IBUF GET THE VALUE BACK TO HL
106B	7E	2579		MOV	A,M
106C	2F	2580		CMA	
106D	77	2581		MOV	M,A TRY TO STORE THERE
106E	BE	2582		CMP	M
106F	C2 00 00	2583		JNZ	START BAD OR MISSING MEMORY
1072	C9	2584		RET	
1073		2585	*		
1073		2586	*	OUTPUT FP NUMBER ADDRESSED BY HL	
1073		2587	*		
1073	01 FC FF	2588	FPOUT	LXI	B,-DIGIT-1
1076	09	2589		DAD	B
1077	44	2590		MOV	B,H
1078	4D	2591		MOV	C,L
1079	21 2F 19	2592		LXI	H,ABUF OUTPUT BUFFER
107C	3A 3A 19	2593		LDA	INFES OUTPUT FORMAT
107F	32 39 19	2594		STA	FES STORE IT
1082	1E 03	2595		MVI	E,DIGIT
1084	36 00	2596		MVI	M,0 CLEAR ROUND-OFF OVERFLOW BUFFER
1086	23	2597		INX	H ABUF+1
1087		2598	*		
1087	0A	2599	NXT	LDAX	B GET DIGIT AND UNPACK
1088	57	2600		MOV	D,A
1089	1F	2601		RAR	
108A	1F	2602		RAR	
108B	1F	2603		RAR	
108C	1F	2604		RAR	
108D	E6 0F	2605		ANI	17Q REMOVE BOTTOM DIGIT
108F	77	2606		MOV	M,A STORE TOP DIGIT IN OUTPUT BUFFER (ABUF)
1090	23	2607		INX	H
1091	7A	2608		MOV	A,D NOW GET BOTTOM DIGIT
1092	E6 0F	2609		ANI	17Q
1094	77	2610		MOV	M,A STORE IT

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

1095 23	2611	INX	H	
1096 03	2612	INX	B	
1097 1D	2613	DCR	E	
1098 C2 87 10	2614	JNZ	NXT	
109B 0A	2615	LDAX	B	
109C 32 29 19	2616	STA	FSIGN	STORE SIGN OF NUMBER
109F AF	2617	XRA	A	
10A0 77	2618	MOV	M,A	CLEAR ROUND-OFF BUFFER (ABUF+13) 12 DIGIT NO RND
10A1 21 37 49	2619	LXI	H,XSIGN	EXPONENT SIGN STORE
10A4 77	2620	MOV	M,A	CLEAR XSIGN
10A5	2621 *			
10A5 03	2622 FIX	INX	B	GET EXPONENT
10A6 0A	2623	LDAX	B	
10A7 B7	2624	ORA	A	EXPONENT ZERO? .
10A8 CA B6 10	2625	JZ	ZRO	
10AB D6 80	2626	SUI	128	REMOVE NORMALIZING BIAS
10AD C2 B1 10	2627	JNZ	FIX2	
10B0 34	2628	INR	M	INCREMENT XSIGN TO NE ATIVE FLAG (1)LATER ZERO
10B1 F2 B7 10	2629 FIX2	JP	CHK43	
10B4 2F	2630	CMA	.	ITS A NEGATIVE EXPONENT
10B5 34	2631	INR	M	INCREMENT XSIGN TO NEGATIVE (1)
10B6 3C	2632 ZRO	INR	A	
10B7 21 38 19	2633 CHK43	LXI	H,EXPO	EXPONENT TEMP STORE
10BA 77	2634	MOV	M,A	
10BB 5F	2635	MOV	E,A	
10BC FE 06	2636	CPI	DIGIT*2	
10BE 21 39 19	2637	LXI	H,FES	FORMAT TEMP BYTE
10C1 DA C8 10	2638	JC	CHKXO	
10C4 3E 01	2639 CHK40	MVI	A,1	FORCE EXPONENTIAL PRINTOUT
10C6 B6	2640	ORA	M SET	FORMAT FOR XOUT
10C7 77	2641	MOV	M,A	
10C8	2642 *			
10C8 7E	2643 CHKXO	MOV	A,M	CHECK IF EXPONENTIAL PRINTOUT
10C9 1F	2644	RAR		
10CA D2 DB 10	2645	JNC	CHKX3	
10CD E6 0F	2646	ANI	17Q	ADD EXPONENT AND DECIMAL PLACES
10CF FE 06	2647	CPI	DIGIT*2	
10D1 DA D6 10	2648	JC	CHKX2	
10D4 3E 05	2649	MVI	A,DIGIT*2-1	MAX DIGITS
10D6 57	2650 CHKX2	MOV	D,A	
10D7 3C	2651	INR	A	
10D8 C3 21 11	2652	JMP	ROUND	
10DB	2653 *			
10DB E6 0F	2654 CHKX3	ANI	17Q	ADD EXPONENT AND DECIMAL PLACES
10DD 57	2655	MOV	D,A	
10DE 83	2656	ADD	E	
10DF FE 07	2657	CPI	DIGIT*2+1	
10E1 47	2658	MOV	B,A	
10E2 DA EB 10	2659	JC	CHKXN	
10E5 7E	2660	MOV	A,M	
10E6 E6 40	2661	ANI	100Q	
10E8 C2 C4 10	2662	JNZ	CHK40	
10EB	2663 *			
10EB 3A 37 19	2664 CHKXN	LDA	XSIGN	CHECK EXPONENT SIGN
10EE B7	2665	ORA	A	
10EF C2 F6 10	2666	JNZ	XNEG	ITS NEGATIVE

10F2 78	2667	MOV	A,B	
10F3 C3 21 11	2668	JMP	ROUND	
10F6	2669	*		
10F6 7A	2670	XNEG MOV	A,D	SUBTRACT EXPONENT AND DECIMAL PLACE COUNT
10F7 93	2671	SUB	E	
10F8 D2 12 11	2672	JNC	XN2	
10FB 3A 3A 19	2673	XN1 LDA	INFES	
10FE B7	2674	ORA	A	
10FF F2 48 12	2675	JP	ZERO	
1102 E6 0E	2676	ANI	16Q	
1104 CA 48 12	2677	JZ	ZERO	
1107 0F	2678	RRC		
1108 5F	2679	MOV	E,A	
1109 1D	2680	DCR	E	
110A 0E 01	2681	MVI	C,1	
110C 21 2E 19	2682	LXI	H,ABUF-1	
110F C3 83 11	2683	JMP	NRND	
1112 CA FB 10	2684	XN2 JZ	XN1	
1115 C3 21 11	2685	JMP	ROUND	
1118	2686	*		
1118	2687	*		
1118 06 1F	2688	CLEAN MVI	B,37Q	CLEAR FLAGS
111A A0	2689	ANA	B	
111B FE 07	2690	CPI	DIGIT*2+1	
111D D8	2691	RC		
111E 3E 07	2692	MVI	A,DIGIT*2+1	MAX DIGITS OUT
1120 C9	2693	RET		
1121	2694	*		
1121	2695	* This routine is used to round data to the		
1121	2696	* specified decimal place		
1121 CD 18 11	2697	ROUND CALL	CLEAN	
1124 4F	2698	MOV	C,A	
1125 06 00	2699	MVI	B,0	
1127 21 30 19	2700	LXI	H,ABUF+1	
112A 09	2701	DAD	B	GET ROUND-OFF ADDRESS
112B 22 22 19	2702	SHLD	ADDT	
112E 7E	2703	MOV	A,M	
112F FE 05	2704	CPI	5	ROUND IF >=5
1131 DA 48 11	2705	JC	TRL2-1	
1134	2706	*		
1134 2B	2707	LESS1 DCX	H	
1135 34	2708	INR	M	ROUND UP
1136 7E	2709	MOV	A,M	
1137 B7	2710	ORA	A	
1138 CA 49 11	2711	JZ	TRL2	
113B FE 0A	2712	CPI	10	CHECK IF ROUNDED NUMBER >9
113D C2 45 11	2713	JNZ	TRAIL	
1140 36 00	2714	MVI	M,0	
1142 C3 34 11	2715	JMP	LESS1	
1145	2716	*		
1145	2717	* This routine is used to eliminate trailing zeros		
1145 2A 22 19	2718	TRAIL LHL	ADDT	
1148 2B	2719	DCX	H	
1149 3A 39 19	2720	TRL2 LDA	FES	CHECK IF TRAILING ZEROS ARE WANTED
114C 17	2721	RAL		
114D DA 5D 11	2722	JC	FPRNT	YES- GO PRINT DATA

1150 7E	2723	TRL3	MOV	A,M	
1151 B7	2724		ORA	A	IS IT A ZERO?
1152 C2 5D 11	2725		JNZ	FPRNT	NO - GO PRINT
1155 2B	2726		DCX	H	
1156 0D	2727		DCR	C	YES- FIX OUTPUT DIGIT COUNT
1157 FA 48 12	2728		JM	ZERO	
115A C3 50 11	2729		JMP	TRL3	
115D	2730	*			
115D	2731	* Here starts the print format routines			
115D 24 2F 19	2732	FPRNT	LXI	H,ABUF	
1160 7E	2733		MOV	A,M	CHECK IF ROUNDED UP TO 1
1161 B7	2734		ORA	A	
1162 CA 83 11	2735		JZ	NRND	JUMP IF NOT
1165 06 01	2736		MVI	B,1	
1167 3A 37 19	2737		LDA	XSIGN	IS EXPONENT NEGATIVE?
116A B7	2738		ORA	A	
116B CA 70 11	2739		JZ	POSR	
116E 06 FF	2740		MVI	B,-1	
1170	2741	*			
1170 3A 38 19	2742	POSR	LDA	EXPO	GET EXPONENT
1173 B7	2743		ORA	A	
1174 C2 7C 11	2744		JNZ	PO2	IS IT ZERO? (E+0)
1177 32 37 19	2745		STA	XSIGN	
117A 06 01	2746		MVI	B,1	
117C 80	2747	PO2	ADD	B	FIX EXPONENT COUNT
117D 32 38 19	2748		STA	EXPO	
1180 1C	2749		INR	E	
1181 0C	2750		INR	C	
1182 2B	2751		DCX	H	
1183	2752	*			
1183 23	2753	NRND	INX	H	
1184 79	2754		MOV	A,C	
1185 FE 07	2755		CPI	DIGIT*2+1	CHECK FOR MAXIMUM DIGITS OUT
1187 C2 8B 11	2756		JNZ	NRND1	
118A 0D	2757		DCR	C	
118B 3A 29 19	2758	NRND1	LDA	FSIGN	CHECK IF NEGATIVE NUMBER
118E 1F	2759		RAR		
118F D2 98 11	2760		JNC	PRIN2	GO OUTPUT RADIX AND NUMBER
1192 CD 43 12	2761		CALL	NEG	OUTPUT (-)
1195 C3 9B 11	2762		JMP	PRI21	
1198	2763	*			
1198 CD 4D 12	2764	PRIN2	CALL	SPACE	OUTPUT A SPACE
119B 3A 39 19	2765	PRI21	LDA	FES	GET OUTPUT FORMAT
119E 1F	2766		RAR	.	CHECK IF EXPONENTIAL OUTPUT FORMAT
119F DA D6 11	2767		JC	XPRIN	
11A2 3A 37 19	2768		LDA	XSIGN	GET EXPONENT SIGN
11A5 B7	2769		ORA	A	CHECK IF NEGATIVE EXPONENT
11A6 CA C8 11	2770		JZ	POSIT	
11A9 79	2771		MOV	A,C	
11AA B7	2772		ORA	A	
11AB C2 B2 11	2773		JNZ	PRIN4	OUTPUT RADIX AND NUMBER
11AE CD 48 12	2774		CALL	ZERO	NO DIGITS AFTER RADIX, OUPUT ZERO AND DONE
11B1 C9	2775		RET		
11B2	2776	*			
11B2 CD 52 12	2777	PRIN4	CALL	RADIX	PRINT DECIMAL POINT
11B5 AF	2778		XRA	A	

11B6 B3	2779	ORA	E	
11B7 CA C1 11	2780	JZ	PRIN5	JUMP IF NO ZEROS TO PRINT
11BA CD 48 12	2781	CALL	ZERO	FORCE PRINT A ZERO
11BD 1D	2782	DCR	E	
11BE C2 B5 11	2783	JNZ	PRIN4+3	
11C1	2784 *			
11C1 CD 39 12	2785	PRIN5 CALL	NOUT	PRINT ASCII DIGIT
11C4 C2 C1 11	2786	JNZ	PRIN5	
11C7 C9	2787	RET		
11C8	2788 *			
11C8 CD 39 12	2789	POSIT CALL	NOUT	
11CB 1D	2790	DCR	E	BUMP EXPONENT COUNT
11CC C2 C8 11	2791	JNZ	POSIT	
11CF 79	2792	MOV	A,C	CHECK IF MORE DIGITS TO OUPUT
11D0 B7	2793	ORA	A	
11D1 C8	2794	RZ	.	NO, DONE
11D2 F8	2795	RM		
11D3 C3 B2 11	2796	JMP	PRIN4	NOW PRINT DECIMAL POINT
11D6	2797 *			
11D6	2798	* Get here for exponential output format		
11D6 CD 39 12	2799	XPRIN CALL	NOUT	
11D9 CA E5 11	2800	JZ	NDEC	INTEGER?
11DC CD 52 12	2801	CALL	RADIX	NO.....PRINT DECIMAL POINT
11DF CD 39 12	2802	XPRI2 CALL	NOUT	
11E2 C2 DF 11	2803	JNZ	XPRI2	
11E5	2804 *			
11E5 06 45	2805	NDEC MVI	B,'E'	OUTPUT 'E'
11E7 CD 95 0E	2806	CALL	CHOUT	
11EA 3A 37 19	2807	LDA	XSIGN	
11ED B7	2808	ORA	A	
11EE CA FB 11	2809	JZ	XPRI3	
11F1 CD 43 12	2810	CALL	NEG	PRINT EXPONENT SIGN (-)
11F4 3A 38 19	2811	LDA	EXPO	
11F7 3C	2812	INR	A	
11F8 C3 04 12	2813	JMP	XOUT2	
11FB 06 2B	2814	XPRI3 MVI	B,'+'	EXPONENT (+)
11FD CD 95 0E	2815	CALL	CHOUT	
1200	2816 *			
1200	2817	* This routine is used to convert the exponent		
1200	2818	* from binary to ASCII and print the result		
1200 3A 38 19	2819	XOUT LDA	EXPO	
1203 3D	2820	DCR	A	
1204 0E 64	2821	XOUT2 MVI	C,100	
1206 16 00	2822	MVI	D,0	
1208 CD 2E 12	2823	CALL	CONV	
120B FE 30	2824	CPI	'0'	SKIP LEADING ZEROS
120D CA 14 12	2825	JZ	X021	
1210 14	2826	INR	D	
1211 CD 95 0E	2827	CALL	CHOUT	
1214 7B	2828	X021 MOV	A,E	
1215 0E 0A	2829	MVI	C,40	
1217 CD 2E 12	2830	CALL	CONV	
121A FE 30	2831	CPI	'0'	
121C C2 23 12	2832	JNZ	X03	
121F 15	2833	DCR	D	
1220 C2 26 12	2834	JNZ	X04	

1223	CD	95	0E	2835	XO3	CALL	CHOUT	
1226	7B			2836	XO4	MOV	A,E	
1227	C6	30		2837		ADI	'0'	ADD ASCII BIAS
1229	47			2838		MOV	B,A	
122A	CD	95	0E	2839		CALL	CHOUT	
122D	C9			2840		RET		
122E	06	2F		2841	CONV	MVI	B,'0'-1	
1230	04			2842		INR	B	
1231	94			2843		SUB	C	
1232	D2	30	12	2844		JNC	CONV+2	
1235	81			2845		ADD	C	
1236	5F			2846		MOV	E,A	
1237	78			2847		MOV	A,B	
1238	C9			2848		RET		
1239				2849	*			
1239				2850	*	This routine adds ASCII bias to a BCD digit		
1239				2851	*	and calls the output routine		
1239	7E			2852	NOUT	MOV	A,M	
123A	C6	30		2853		ADI	'0'	
123C	47			2854		MOV	B,A	
123D	CD	95	0E	2855		CALL	CHOUT	
1240	23			2856		INX	H	
1241	0D			2857		DCR	C	DECREMENT TOTAL DIGITS OUT COUNT
1242	C9			2858		RET		
1243				2859	*			
1243				2860	*	Common symbol loading routines		
1243	06	2D		2861	NEG	MVI	B,'-'	
1245	C3	95	0E	2862		JMP	CHOUT	
1248	06	30		2863	ZERO	MVI	B,'0'	
124A	C3	95	0E	2864		JMP	CHOUT	
124D	06	20		2865	SPACE	MVI	B,' '	
124F	C3	95	0E	2866		JMP	CHOUT	
1252	06	2E		2867	RADIX	MVI	B,'.'	
1254	C3	95	0E	2868		JMP	CHOUT	
1257				2869	*	CONVERTS FP STRING AT DE, UPDATE DE PAST TERMINATOR		
1257				2870	*	PUTS TERMINATOR IN B, PUTS FP NUMBER AT ADDRESS IN HL		
1257				2871	*	SETS CARRY IF NOT FOUND		
1257	E5			2872	FPIN	PUSH	H	
1258	D5			2873		PUSH	D	
1259	EB			2874		XCHG		
125A	2B			2875		DCX	H	
125B	22	20	19	2876		SHLD	ADDS	
125E	24	26	19	2877		LXI	H,OPST	CLEAR TEMPORARY STORAGE AREAS AND BC BUFFER
1261	0E	09		2878		MVI	C,DIGIT+6	
1263	CD	1B	13	2879		CALL	CLEAR	
1266				2880	*			
1266	41	00	00	2881	SCANC	LXI	D,0	
1269	24	2A	19	2882		LXI	H,BC	BC=PACK BUFFER
126C	22	24	19	2883	SCANO	SHLD	BCADD	PACK BUFFER POINTER
126F	24	6F	12	2884	SCANP	LXI	H,SCANP	
1272	E5			2885		PUSH	H	USED FOR RETURN FROM OTHER ROUTINES
1273	AF			2886		XRA	A	
1274	32	37	19	2887		STA	XSIGN	CLEAR EXPONENT SIGN BYTE
1277				2888	*			
1277	CD	ED	12	2889	SCANG	CALL	IBSCN	
127A	DA	A4	12	2890		JC	SCANX	FOUND A NUMBER, GO PACK IT

127D FE 2E	2891	CPI	'.'	RADIX?
127F CA 95 12	2892	JZ	SCAN5	PROCESS RADIX POINTERS
1282 FE 45	2893	CPI	'E'	EXPONENT?
1284 CA 23 13	2894	JZ	EXCON	FOUND 'E', GO PROCESS EXPONENT NUMBER
1287	2895	*NOT A CHARACTER LEGAL IN NUMBER		
1287 47	2896	MOV	B,A	MOVE TERMINATOR TO B
1288 3A 26 19	2897	LDA	OPST	CHECK IF ANY DIGITS YET
128B E6 10	2898	ANI	20Q	
128D C2 01 13	2899	JNZ	ENTR2	
1290	2900	*GET HERE IF LEGAL FP NUMBER NOT FOUND		
1290 E1	2901	FPIN1	POP	H SCANP LINK
1291 D1	2902	POP	D	TEXT POINTER
1292 E1	2903	POP	H	FP # ADDR
1293 37	2904	STC		
1294 C9	2905	RET		
1295	2906	*FOUND DECIMAL POINT		
1295 AF	2907	SCAN5	XRA	A FOUND RADIX PROCESS RADIX POINTERS FOR EXP
1296 B2	2908	ORA	D	ANY DIGITS YET?
1297 C2 9F 12	2909	JNZ	SCAN6	
129A C6 C0	2910	ADI	300Q	SET ECNT - STOP COUNTING DIGITS
129C B3	2911	ORA	E	NO INT DIGITS, BIT 7 IS COUNT/DON'T COUNT FLAG
129D 5F	2912	MOV	E,A	BIT 6 IS NEGATIVE EXPONENT FLAG
129E C9	2913	RET		
129F 3E 80	2914	SCAN6	MVI	A,200Q SET ECNT TO COUNT DIGITS
12A1 B3	2915	ORA	E	
12A2 5F	2916	MOV	E,A	
12A3 C9	2917	RET		
12A4	2918	*		
12A4 E6 0F	2919	SCANX	ANI	17Q FOUND NUMBER-REMOVE ASCII BIAS
12A6 47	2920	MOV	B,A	
12A7 21 26 19	2921	LXI	H,OPST	SET FIRST CHARACTER FLAG
12AA 3E 30	2922	MVI	A,60Q	
12AC B6	2923	ORA	M	
12AD 77	2924	MOV	M,A	
12AE AF	2925	XRA	A	
12AF B0	2926	ORA	B	IS CHARACTER ZERO?
12B0 C2 BC 12	2927	JNZ	PACK	
12B3 B2	2928	ORA	D	LEADING ZERO? IE; ANY INT DIGITS?
12B4 C2 BC 12	2929	JNZ	PACK	
12B7 B3	2930	ORA	E	
12B8 5F	2931	MOV	E,A	
12B9 C8	2932	RZ	.	IF COUNTING YET,
12BA 1C	2933	INR	E	ECNT+1-COUNT ZEROS FOR EXPONENT COUNT
12BB C9	2934	RET		
12BC	2935	*		
12BC	2936	* This subroutine BCD packs digits into reg BC		
12BC	2937	*		
12BC 7B	2938	PACK	MOV	A,E
12BD 17	2939	RAL		
12BE DA C2 12	2940	JC	PACK1	
12C1 1C	2941	INR	E	
12C2 7B	2942	PACK1	MOV	A,E
12C3 32 28 19	2943	STA	ECNT	DIGIT COUNT FOR EXPONENT COUNT
12C6 14	2944	INR	D	TOTAL DIGIT COUNT (D ALSO HAS TOP/BOTM FLAG BIT-7)
12C7 7A	2945	MOV	A,D	
12C8 E6 7F	2946	ANI	177Q	REMOVE TOP/BOTTOM FLAG

12CA FE 07	2947	CPI	DIGIT*2+1	LIMIT INPUT DIGITS
12CC D0	2948	RNC		
12CD AF	2949	XRA	A	
12CE B2	2950	ORA	D	
12CF FA DF 12	2951	JM	BOTM	
12D2	2952	*		
12D2 F6 80	2953	TOP ORI	200Q	SET MSB FOR TOP FLAG
12D4 57	2954	MOV	D,A	
12D5 78	2955	MOV	A,B	
12D6 2A 24 19	2956	LHLD	BCADD	GET BC ADDRESS
12D9 07	2957	RLC		
12DA 07	2958	RLC		
12DB 07	2959	RLC		
12DC 07	2960	RLC		
12DD 77	2961	MOV	M,A	SAVE CHR IN BC
12DE C9	2962	RET		
12DF	2963	*		
12DF E6 7F	2964	BOTM ANI	177Q	STRIP MSB (BOTTOM FLAG)
12E1 57	2965	MOV	D,A	
12E2 78	2966	MOV	A,B	
12E3 2A 24 19	2967	LHLD	BCADD	
12E6 B6	2968	ORA	M	OR IN TOP NUMBER
12E7 77	2969	MOV	M,A	PUT NUMBER BACK IN BC
12E8 23	2970	INX	H	
12E9 C1	2971	POP	B	
12EA C3 6C 12	2972	JMP	SCANO	
12ED 2A 20 19	2973	IBSCN LHLD	ADDS	INPUT BUFFER POINTER
12F0 23	2974	INX	H	GET NEXT BYTE
12F1 7E	2975	MOV	A,M	
12F2 FE 20	2976	CPI	' '	
12F4 CA F0 12	2977	JZ	IBSCN+3	
12F7 22 20 19	2978	SHLD	ADDS	NOTE: THIS ROUTINE FALLS THROUGH TO BELOW
12FA	2979	*THIS ROUTINE CHECKS FOR ASCII NUMBERS (0-9)		
12FA FE 3A	2980	NMCHK CPI	'9'+1	
12FC D0	2981	RNC		
12FD FE 30	2982	CPI	'0'	
12FF 3F	2983	CMC		
1300 C9	2984	RET		
1301	2985	*		
1301	2986	* This routine is used to adjust a number in BC BUFFER		
1301	2987	* AND RETURNS VALUE		
1301 11 00 00	2988	ENTR2 LXI	D,0	
1304 C5	2989	ENT1 PUSH	B	TERMINATOR
1305 CD 7E 13	2990	CALL	FIXE	NORMALIZE FLOATING POINT NUMBER
1308 C1	2991	POP	B	TERMINATOR
1309 D1	2992	POP	D	SCANP LINK
130A D1	2993	POP	D	OLD TEXT ADDR
130B D1	2994	POP	D	RETURN ADDR
130C 0E 05	2995	MVI	C,DIGIT+2	
130E 21 2E 19	2996	LXI	H,BC+DIGIT+1	
1311 CD 9C 0C	2997	CALL	VCOPY	
1314 2A 20 19	2998	LHLD	ADDS	
1317 EB	2999	XCHG		
1318 13	3000	INX	D	
1319 B7	3001	ORA	A	
131A C9	3002	RET		

131B	3003	* This routine is used to clear storage areas	
131B	3004	* The starting address is in H&L and the count	
131B	3005	* is in reg C	
131B AF	3006	CLEAR	XRA A
131C 77	3007		MOV M,A
131D 23	3008		INX H
131E 0D	3009		DCR C
131F C2 1C 13	3010		JNZ CLEAR+1
1322 C9	3011		RET
1323	3012	*	
1323	3013	* This routine converts the ASCII exponent of	
1323	3014	* number in the input buffer to binary, and	
1323	3015	* normalizes exponent according to the input	
1323	3016	* format of the number	
1323 CD ED 12	3017	EXCON	CALL IBSCN GET CHARACTER
1326 DA 48 13	3018		JC EXC3
1329 FE E3	3019		CPI PLSRW CHECK FOR UNARY SIGNS
132B CA 42 13	3020		JZ EXC4
132E FE 2B	3021		CPI '+'
1330 CA 42 13	3022		JZ EXC4
1333 FE 2D	3023		CPI '-'
1335 CA 3D 13	3024		JZ EXC2
1338 FE E5	3025		CPI MINRW
133A C2 7A 13	3026		JNZ FPERR NO SIGN OR NUMBER?
133D 3E 01	3027	EXC2	MVI A,1
133F 32 37 19	3028		STA XSIGN SAVE SIGN
1342 CD ED 12	3029	EXC4	CALL IBSCN
1345 D2 7A 13	3030		JNC FPERR NO NUMBER?
1348 CD 4E 13	3031	EXC3	CALL ASCDC CONVERT ASCII TO BINARY
134B C3 04 13	3032		JMP ENT1 NORMALIZE NUMBER AND RETURN
134E	3033	*	
134E	3034	* This routine converts ASCII to binary	
134E	3035	* Three consecutive numbers <128 may be converted	
134E EB	3036	ASCDC	XCHG
134F 21 00 00	3037		LXI H,0
1352 1A	3038	ASC1	LDAX D GET CHR FROM INPUT BUFFER- NO SPACES ALLOWED
1353 CD FA 12	3039		CALL NMCHK CHECK IF NUMBER
1356 D2 69 13	3040		JNC ASC2
1359 D6 30	3041		SUI '0' REMOVE ASCII BIAS
135B 44	3042		MOV B,H
135C 4D	3043		MOV C,L
135D 29	3044		DAD H
135E 29	3045		DAD H
135F 09	3046		DAD B
1360 29	3047		DAD H
1361 4F	3048		MOV C,A
1362 06 00	3049		MVI B,0
1364 09	3050		DAD B
1365 13	3051		INX D
1366 C3 52 13	3052		JMP ASC1
1369 EB	3053	ASC2	XCHG
136A 47	3054		MOV B,A SAVE TERMINATOR
136B 22 20 19	3055		SHLD ADDS SAVE IBUF ADDRESS
136E 7A	3056		MOV A,D
136F B7	3057		ORA A
1370 C2 7A 13	3058		JNZ FPERR TCO BIG >255

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

1373 7B	3059	MOV	A,E	
1374 17	3060	RAL		
1375 DA 7A 13	3061	JC	FPERR	TOO BIG >127
1378 1F	3062	RAR		
1379 C9	3063	RET		
137A C1	3064	FPERR POP	B	ASCDC RET LINK
137B C3 90 12	3065	JMP	FPIN4	
137E	3066	*		
137E	3067	* This routine normalizes the input number		
137E EB	3068	FIXE XCHG		
137F 3A 2A 19	3069	LDA	BC	
1382 B7	3070	ORA	A	IS IT ZERO?
1383 CA 8B 13	3074	JZ	ZZ2	
1386 CD 8F 13	3072	CALL	CHKPN	SET EXPONENT POSITIVE/NEGATIVE
1389 C6 80	3073	ADI	200Q	ADD EXPONENT BIAS
138B 32 2E 19	3074	ZZ2 STA	BC+DIGIT+1	STORE NORMALIZED EXPONENT IN BC
138E C9	3075	RET		
138F	3076	*		
138F 3A 28 19	3077	CHKPN LDA	ECNT	GET EXPONENT COUNT-SET IN 'SCAN' ROUTINE
1392 5F	3078	MOV	E,A	
1393 E6 3F	3079	ANI	77Q	STRIP BITS 7&8
1395 47	3080	MOV	B,A	
1396 3A 37 19	3081	LDA	XSIGN	
1399 B7	3082	ORA	A	
139A CA B1 13	3083	JZ	LPOS	EXPONENT IS POSITIVE
139D 24	3084	INR	H	SET SIGN IN H ** THIS SHOULD BE INR H NOT INX H
139E 3E 40	3085	MVI	A,100Q	L IS NEGATIVE
13A0 A3	3086	ANA	E	CHECK IF E IS NEGATIVE
13A1 CA AC 13	3087	JZ	EPOS	
13A4 7D	3088	MOV	A,L	BOTH E&L NEGATIVE
13A5 68	3089	MOV	L,B	
13A6 CD BD 13	3090	CALL	BPOS+1	
13A9 2F	3091	CMA		
13AA 3C	3092	INR	A	
13AB C9	3093	RET	.	BACK TO FIXE
13AC	3094	*		
13AC 7D	3095	EPOS MOV	A,L	E&L NEGATIVE
13AD 2F	3096	CMA		
13AE 3C	3097	INR	A	
13AF 80	3098	ADD	B	
13B0 C9	3099	RET	.	TO FIXE
13B1	3100	*		
13B1 3E 40	3101	LPOS MVI	A,100Q	EXPONENT POSITIVE
13B3 A3	3102	ANA	E	IS E NEGATIVE?
13B4 CA BC 13	3103	JZ	BPOS	
13B7 78	3104	MOV	A,B	
13B8 45	3105	MOV	B,L	
13B9 C3 AD 13	3106	JMP	EPOS+1	
13BC	3107	*		
13BC 78	3108	BPOS MOV	A,B	E&L POSITIVE
13BD 85	3109	ADD	L	
13BE F0	3110	RP		
13BF	3111	*		
13BF E1	3112	POP	H	
13C0 C3 7A 13	3113	JMP	FPERR	
13C3 10	3114	DB	1*16	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

13C4 00 00	3115	DW	0	
13C6 01	3116	DB	1	
13C7 81	3117	FPNONE	DB	129
13C8	3118	*		
13C8	3119	*	THIS PROGRAM WRITTEN BY	
13C8	3120	*	MICROTEC	
13C8	3121	*	990 E. ARQUES	
13C8	3122	*	SUNNYVALE, CA. 94006	
13C8	3123	*		
13C8	3124	*	THIS PROGRAM IS A FOUR FUNCTION FLOATING POINT BCD	
13C8	3125	*	MATH PACKAGE.	
13C8	3126	*	EACH FUNCTION MAY BE EXPRESSED AS: BC=DE # HL	
13C8	3127	*	<BC> = ADDRESS OF RESULT	
13C8	3128	*	<DE> = ADDRESS OF 1ST ARGUMENT	
13C8	3129	*	<HL> = ADDRESS OF 2ND ARGUMENT	
13C8	3130	*	# IS ONE OF THE FUNCTIONS: +, -, X, /.	
13C8	3131	*	ALL ADDRESSES ON ENTRY, POINT TO THE EXPONENT PART OF	
13C8	3132	*	THE FLOATING POINT NUMBER.	
13C8	3133	*	EACH FLOATING POINT NUMBER CONSISTS OF (2*DIGIT) PACKED	
13C8	3134	*	DECIMAL DIGITS, A SIGN AND A BIASED BINARY EXPONENT. THE	
13C8	3135	*	EXPONENT RANGE IS 10** -127 TO 10**127.	
13C8	3136	*	THE NUMBER ZERO IS REPRESENTED BY THE EXPONENT 0.	
13C8	3137	*	THE NUMBERS ARE STORED IN MEMORY AS (DIGIT) BYTES OF	
13C8	3138	*	OF DECIMAL DIGITS	
13C8	3139	*	STARTING AT THE LOW ORDER ADDRESS	
13C8	3140	*	ALL NUMBER ARE ASSUMED TO BE NORMALIZED. THAT IS EACH	
13C8	3141	*	NUMBER CAN BE REPRESENTED AS F**E.	
13C8	3142	*	WHERE .1<=F<4.0 AND F IS THE	
13C8	3143	*	EXPONENT.	
13C8	3144	*		
13C8	3145	*	FLOATING POINT ADDITION	
13C8	3146	*		
13C8 C5	3147	FADD	PUSH	B
13C9 CD 42 16	3148		CALL	EXPCK
13CC 0E 00	3149		MVI	C,0
13CE 1B	3150	ADSUM	DCX	D
13CF EB	3151		XCHG	
13D0 3A 28 17	3152		LDA	SIGN
13D3 AE	3153		XRA	M
13D4 47	3154		MOV	B,A
13D5 EB	3155		XCHG	
13D6 1A	3156		LDAX	D
13D7 1B	3157		DCX	D
13D8 A9	3158		XRA	C
13D9 32 28 17	3159		STA	SIGN
13DC 21 2A 17	3160		LXI	H,RCTRL
13DF 7E	3161		MOV	A,M
13E0 B7	3162		ORA	A
13E1 23	3163		INX	H
13E2 7E	3164		MOV	A,M
13E3 CA EA 13	3165		JZ	ADS8
13E6 07	3166		RLC	
13E7 07	3167		RLC	
13E8 07	3168		RLC	
13E9 07	3169		RLC	
13EA C6 B0	3170	ADS8	ADI	OBOH

FORCE CARRY IF DIGIT > 5

13EC 78	3174	MOV	A,B	
13ED 1F	3172	RAR		
13EE DA 2E 14	3173	JC	ADS4	HAVE SUBTRACTION
13F1 17	3174	RAL	.	RESTORE CARRY
13F2 CD OD 14	3175	CALL	ADD	PERFORM ADDITION
13F5 D2 04 14	3176	JNC	ADS2	
13F8 06 04	3177	MVI	B,4	
13FA CD A8 16	3178	CALL	RIGHT	
13FD 21 29 17	3179	LXI	H,EXP	
1400 34	3180	INR	M	INCREMENT EXPONENT
1401 CA F4 16	3181	JZ	OVER	
1404 C1	3182	POP	B	GET RESULTS ADDRESS
1405 CD 9A 16	3183	CALL	STORE	SAVE RESULTS
1408 C9	3184	RET		
1409 E1	3185	ZEREX	POP	H
140A C3 04 14	3186	JMP	ADS2	
140D 21 27 17	3187	ADD	LXI	H,BUF+DIGIT-1
1410 06 03	3188	MVI	B,DIGIT	
1412 1A	3189	ADD4	LDAX	D
1413 8E	3190	ADC	M	
1414 27	3191	DAA		
1415 77	3192	MOV	M,A	
1416 2B	3193	DCX	H	
1417 1B	3194	DCX	D	
1418 05	3195	DCR	B	
1419 C2 12 14	3196	JNZ	ADD4	
141C D0	3197	RNC		
141D 34	3198	INR	M	
141E C9	3199	RET		
141F	3200	*		
141F	3201	*	FLOATING POINT SUBTRACTION	
141F	3202	*		
141F C5	3203	FSUB	PUSH	B
1420 CD 42 16	3204	CALL	EXPCN	GET ARGUMENTS
1423 3A 28 17	3205	LDA	SIGN	
1426 EE 01	3206	XRI	1	COMPLEMENT SIGN
1428 32 28 17	3207	STA	SIGN	
142B C3 CE 13	3208	JMP	ADSUM	
142E 17	3209	ADS4	RAL	RESTORE CARRY
142F 3F	3210	CMC	.	COMPLEMENT FOR ROUNDING
1430 CD 82 14	3211	CALL	SUB	SUBTRACT ARGUMENTS
1433 21 28 17	3212	LXI	H,SIGN	
1436 DA 4D 14	3213	JC	ADS4	
1439 7E	3214	MOV	A,M	GET SIGN
143A EE 01	3215	XRI	1	COMPLEMENT
143C 77	3216	MOV	M,A	
143D 2B	3217	ADS7	DCX	H
143E 06 03	3218	MVI	B,DIGIT	
1440 3E 9A	3219	ADS3	MVI	A,9AH
1442 9E	3220	SBB	M	COMPLEMENT RESULT
1443 C6 00	3221	ADI	0	
1445 27	3222	DAA		
1446 77	3223	MOV	M,A	
1447 2B	3224	DCX	H	
1448 05	3225	DCR	B	
1449 3F	3226	CMC		

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

144A	C2 40 14	3227	JNZ	ADS3	
144D	21 25 17	3228	ADSS	LXI	H,BUF
1450	01 03 00	3229		LXI	B,DIGIT
1453	7E	3230	ADSS	MOV	A,M
1454	B7	3231		ORA	A
1455	C2 66 14	3232		JNZ	ADS6
1458	23	3233		INX	H
1459	04	3234		INR	B
145A	04	3235		INR	B
145B	0D	3236		DCR	C
145C	C2 53 14	3237		JNZ	ADS5
145F	AF	3238		XRA	A
1460	32 29 17	3239		STA	EXP
1463	C3 04 14	3240		JMP	ADS2
1466	FE 10	3241	ADSS	CPI	#0H
1468	D2 6C 14	3242		JNC	ADS9
146B	04	3243		INR	B
146C	21 29 17	3244	ADSS	LXI	H,EXP
146F	7E	3245		MOV	A,M
1470	90	3246		SUB	B
1471	CA FA 16	3247		JZ	UNDER
1474	DA FA 16	3248		JC	UNDER
1477	77	3249		MOV	M,A
1478	78	3250		MOV	A,B
1479	07	3251		RLC	
147A	07	3252		RLC	
147B	47	3253		MOV	B,A
147C	CD CE 16	3254		CALL	LEFT
147F	C3 04 14	3255		JMP	ADS2
1482	21 27 17	3256	SUB	LXI	H,BUF+DIGIT-1
1485	06 03	3257		MVI	B,DIGIT
1487	3E 99	3258	SUB1	MVI	A,99H
1489	CE 00	3259		ACI	0
148B	96	3260		SUB	M
148C	EB	3261		XCHG	
148D	86	3262		ADD	M
148E	27	3263		DAA	
148F	EB	3264		XCHG	
1490	77	3265		MOV	M,A
1491	2B	3266		DCX	H
1492	1B	3267		DCX	D
1493	05	3268		DCR	B
1494	C2 87 14	3269		JNZ	SUB1
1497	C9	3270		RET	
1498		3271	*		
1498		3272	*	FLOATING POINT MULTIPLY	
1498		3273	*		
1498	C5	3274	FMUL	PUSH	B
1499	7E	3275		MOV	A,M
149A	B7	3276		ORA	A
149B	CA B2 14	3277		JZ	FMUL1+2
149E	1A	3278		LDAX	D
149F	B7	3279		ORA	A
14A0	CA B2 14	3280		JZ	FMUL1+2
14A3	86	3281		ADD	M
14A4	DA AD 14	3282		JC	FMOVR

ARGUMENT = 0?

ARGUMENT = 0?

FORM RESULT EXPONENT

14A7	F2 FA 16	3283	JP	UNDER	
14AA	C3 B0 14	3284	JMP	FMUL1	
14AD	FA F4 16	3285	FMOVR	JM	OVER
14B0	D6 80	3286	FMUL1	SUI	428 REMOVE EXCESS BIAS
14B2	32 29 17	3287		STA	EXP SAVE EXPONENT
14B5	1B	3288		DCX	D
14B6	2B	3289		DCX	H
14B7	1A	3290		LDAX	D
14B8	AE	3291		XRA	M FORM RESULT SIGH
14B9	2B	3292		DCX	H
14BA	1B	3293		DCX	D
14BB	E5	3294		PUSH	H
14BC	24 28 17	3295		LXI	H,SIGN GET SIGN ADDRESS
14BF	77	3296		MOV	M,A SAVE SIGN
14C0	2B	3297		DCX	H
14C1	AF	3298		XRA	A
14C2	06 05	3299		MVI	B,DIGIT+2
14C4	77	3300	FMUL2	MOV	M,A ZERO WORKING BUFFER
14C5	2B	3301		DCX	H
14C6	05	3302		DCR	B
14C7	C2 C4 14	3303		JNZ	FMUL2
14CA	3A 29 17	3304		LDA	EXP
14CD	B7	3305		ORA	A
14CE	CA 09 14	3306		JZ	ZEREX
14D1	0E 03	3307		MVI	C,DIGIT
14D3	24 05 17	3308		LXI	H,HOLD1+DIGIT
14D6		3309	* GET MULTIPLIER INTO HOLDING REGISTER		
14D6	1A	3310	FMUL3	LDAX	D
14D7	77	3311		MOV	M,A PUT IN REGISTER
14D8	2B	3312		DCX	H
14D9	1B	3313		DCX	D
14DA	0D	3314		DCR	C
14DB	C2 D6 14	3315		JNZ	FMUL3
14DE	74	3316		MOV	M,C
14DF	2B	3317		DCX	H
14E0	06 FA	3318		MVI	B,250 SET LOOP COUNT
14E2	11 04 00	3319	FMUL4	LXI	D,DIGIT+4
14E5	4B	3320		MOV	C,E
14E6	19	3321		DAD	D
14E7	EB	3322		XCHG	
14E8	19	3323		DAD	D H,L=NEXT HOLDING REGISTER
14E9	04	3324		INR	B
14EA	F2 1E 15	3325		JP	FMUL8 FINISHED
14ED	1A	3326	FMUL5	LDAX	D GET DIGITS
14EE	8F	3327		ADC	A TIMES 2
14EF	27	3328		DAA	
14F0	77	3329		MOV	M,A PUT IN HOLDING REGISTER
14F1	1B	3330		DCX	D
14F2	2B	3331		DCX	H
14F3	0D	3332		DCR	C
14F4	C2 ED 14	3333		JNZ	FMUL5
14F7	04	3334		INR	B INCRREMENT LOOP COUNT
14F8	C2 E2 14	3335		JNZ	FMUL4
14FB		3336	* FORM 10X BY ADDING 8X AND 2X		
14FB		3337	* FIRST GET 8X		
14FB	23	3338		INX	H

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

14FC 11 12 17	3339	LXI	D,HOLD5	NEXT HOLDING REGISTER
14FF 0E 04	3340	MVI	C,DIGIT+1	
1501 41	3341	MOV	B,C	
1502 7E	3342	MOV	A,M	
1503 12	3343	STAX	D	
1504 23	3344	INX	H	
1505 13	3345	INX	D	
1506 0D	3346	DCR	C	
1507 C2 02 15	3347	JNZ	FMUL6	
150A 21 09 17	3348	LXI	H,HOLD2+DIGIT	GET 2X
150D 1B	3349	DCX	D	
150E 1A	3350	LDAX	D	
150F 8E	3351	ADC	M	FORM 10X
1510 27	3352	DAA		
1511 12	3353	STAX	D	
1512 1B	3354	DCX	D	
1513 2B	3355	DCX	H	
1514 05	3356	DCR	B	
1515 C2 0E 15	3357	JNZ	FMUL7	
1518 06 F9	3358	MVI	B,249	
151A EB	3359	XCHG		
151B C3 E2 14	3360	JMP	FMUL4	
151E EB	3361	XCHG		
151F 23	3362	INX	H	
1520 36 04	3363	MVI	M,DIGIT+1	SET NEXT LOOP COUNT
1522	3364	* PERFORM ACCUMULATION OF PRODUCT		
1522 C1	3365	FMUL9	POP	B GET MULTIPLIER
1523 21 22 17	3366	LXI	H,HOLD8+DIGIT+1	
1526 35	3367	DCR	M	DECREMNT LOOP COUNT
1527 CA 54 15	3368	JZ	FMU14	FINISHED
152A 0A	3369	LDAX	B	
152B 0B	3370	DCX	B	
152C C5	3371	PUSH	B	
152D 2B	3372	DCX	H	
152E EB	3373	XCHG		
152F 87	3374	FMU10	ADD	A CHECK FOR BIT IN CARRY
1530 DA 3E 15	3375	JC	FMU11	FOUND A BIT
1533 CA 4C 15	3376	JZ	FMU12	ZERO - FINISHED THIS DIGIT
1536 21 FC FF	3377	LXI	H,-DIGIT-1	
1539 19	3378	DAD	D	POINT TO NEXT HOLDING REGISTER
153A EB	3379	XCHG		
153B C3 2F 15	3380	JMP	FMU10	
153E 4F	3381	FMU11	MOV	C,A
153F B7	3382	ORA	A	CLEAR CARRY
1540 CD 0D 14	3383	CALL	ADD	ACCUMULATE PRODUCT
1543 1A	3384	LDAX	D	
1544 86	3385	ADD	M	
1545 27	3386	DAA		
1546 77	3387	MOV	M,A	
1547 79	3388	MOV	A,C	
1548 1B	3389	DCX	D	
1549 C3 2F 15	3390	JMP	FMU10	
154C	3391	* ROTATE RIGHT 1 BYTE		
154C 06 08	3392	FMU12	MVI	B,8
154E CD A8 16	3393	CALL	RIGHT	
1551 C3 22 15	3394	JMP	FMUL9	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

1554 3A 25 17	3395 FMU14	LDA	BUF	
1557 E6 F0	3396	ANI	OFOH	CHECK IF NORMALIZED
1559 CA 65 15	3397	JZ	FMU17	
155C 7A	3398	MOV	A,D	
155D E6 F0	3399	ANI	OFOH	
155F 24 27 17	3400	LXI	H,SIGN-1	
1562 C3 76 15	3401	JMP	FMU18	
1565 06 04	3402 FMU17	MVI	B,4	
1567 24 29 17	3403	LXI	H,EXP	
156A 35	3404	DCR	M	
156B CA FA 16	3405	JZ	UNDER	
156E CD CE 16	3406	CALL	LEFT	NORMALIZE
1571 7A	3407	MOV	A,D	GET DIGIT SHIFTED OFF
1572	3408	* PERFORM ROUNDING		
1572 OF	3409	RRC		
1573 OF	3410	RRC		
1574 OF	3411	RRC		
1575 OF	3412	RRC		
1576 FE 50	3413 FMU18	CPI	50H	
1578 DA 9A 15	3414	JC	FMU16	
157B 3C	3415	INR	A	
157C E6 0F	3416	ANI	OFOH	
157E 0E 03	3417	MVI	C,DIGIT	
1580 8E	3418 FMU15	ADC	M	
1581 27	3419	DAA		
1582 77	3420	MOV	M,A	
1583 3E 00	3421	MVI	A,0	
1585 2B	3422	DCX	H	
1586 0D	3423	DCR	C	
1587 C2 80 15	3424	JNZ	FMU15	
158A	3425	* CHECK FOR ROUNDING OVERFLOW		
158A D2 04 14	3426	JNC	ADS2	NO OVERFLOW
158D 23	3427	INX	H	
158E 36 10	3428	MVI	M,10H	
1590 24 29 17	3429	LXI	H,EXP	
1593 34	3430	INR	M	
1594 C2 04 14	3431	JNZ	ADS2	
1597 C3 F4 16	3432	JMP	OVER	
159A	3433	* ROUNDING NOT NEEDED		
159A E6 0F	3434 FMU16	ANI	OFOH	
159C 86	3435	ADD	M	
159D 77	3436	MOV	M,A	
159E C3 04 14	3437	JMP	ADS2	
15A1	3438	*		
15A1	3439	* FLOATING POINT DIVISION		
15A1	3440	*		
15A1 C5	3441 FDIV	PUSH	B	
15A2 7E	3442	MOV	A,M	FETCH DIVISOR EXP
15A3 B7	3443	ORA	A	DIVIDE BY 0?
15A4 CA F4 16	3444	JZ	DIVZ	
15A7 1A	3445	LDAX	D	
15A8 B7	3446	ORA	A	DIVIDEND = 0?
15A9 CA FF 16	3447	JZ	INSP	
15AC 96	3448	SUB	M	
15AD DA B6 15	3449	JC	DIVUN	
15B0 FA F4 16	3450	JM	OVER	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

15B3 C3 B9 15	3451	JMP	FDI4	
15B6 F2 FA 16	3452	DIVUN	JP	UNDER
15B9 C6 81	3453	FDI4	ADI	429 FORM_QUOTIENT EXP
15BB 32 06 17	3454		STA	EXPD
15BE EB	3455	XCHG		
15BF D5	3456	PUSH	D	
15C0 CD 84 16	3457	CALL	LOAD	FETCH DIVIDEND
15C3 D4	3458	POP	D	
15C4 EB	3459	XCHG		
15C5 3A 28 17	3460	LDA	SIGN	
15C8 2B	3461	DCX	H	
15C9 AE	3462	XRA	M	FORM QUOTIENT SIGN
15CA 32 05 17	3463	STA	SIGND	
15CD EB	3464	XCHG		
15CE 1B	3465	DCX	D	
15CF 01 02 17	3466	LXI	B,HOLD4	
15D2 2E 06	3467	DIVO	MVI	L,DIGIT+DIGIT
15D4 C5	3468	DIV4	PUSH	B
15D5 E5	3469		PUSH	H
15D6 9E 00	3470	MVI	C,0	QUOTIENT DIGIT = 0
15D8 37	3471	DIV3	STC	SET CARRY
15D9 24 27 17	3472		LXI	H,BUF+DIGIT-1
15DC 06 03	3473		MVI	B,DIGIT
15DE 3E 99	3474	DIV4	MVI	A,99H
15E0 CE 00	3475		ACI	0
15E2 EB	3476	XCHG		
15E3 96	3477	SUB	M	
15E4 EB	3478	XCHG		
15E5 86	3479	ADD	M	
15E6 27	3480	DAA		
15E7 77	3481	MOV	M,A	
15E8 2B	3482	DCX	H	
15E9 1B	3483	DCX	D	
15EA 05	3484	DCR	B	
15EB C2 DE 15	3485	JNZ	DIV4	
15EE 7E	3486	MOV	A,M	
15EF 3F	3487	CMC		
15F0 DE 00	3488	SBI	0	
15F2 77	3489	MOV	M,A	
15F3 1F	3490	RAR		
15F4 24 03 00	3491	LXI	H,DIGIT	
15F7 19	3492	DAD	D	
15F8 EB	3493	XCHG		
15F9 0C	3494	INR	C	INCREMENT QUOTIENT
15FA 17	3495	RAL		
15FB D2 D8 15	3496	JNC	DIV3	
15FE B7	3497	ORA	A	CLEAR CARRY
15FF CD 0D 14	3498	CALL	ADD	RESTORE DIVIDEND
1602 24 03 00	3499	LXI	H,DIGIT	
1605 19	3500	DAD	D	
1606 EB	3501	XCHG		
1607 C5	3502	PUSH	B	
1608 06 04	3503	MVI	B,4	
160A CD CE 16	3504	CALL	LEFT	SHIFT DIVIDEND
160D C1	3505	POP	B	
160E 0D	3506	DCR	C	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

160F E1	3507	POP	H	
1610 61	3508	MOV	H,C	
1611 C1	3509	POP	B	
1612 7D	3510	MOV	A,L	
1613 C2 25 16	3511	JNZ	DIV5	
1616 FE 06	3512	CPI	DIGIT+DIGIT	
1618 C2 25 16	3513	JNZ	DIV5	
161B 21 06 17	3514	LXI	H,EXPD	
161E 35	3515	DCR	M	
161F CC FA 16	3516	CZ	UNDER	
1622 C3 D2 15	3517	JMP	DIV0	
1625 1F	3518	RAR		
1626 7C	3519	MOV	A,H	
1627 D2 35 16	3520	JNC	DIV6	
162A 0A	3521	LDAX	B	
162B 07	3522	RLC		
162C 07	3523	RLC		
162D 07	3524	RLC		
162E 07	3525	RLC		
162F 84	3526	ADD	H	
1630 02	3527	STAX	B	STORE QUOTIEN
1631 03	3528	INX	B	
1632 C3 36 16	3529	JMP	DIV7	
1635 02	3530	STAX	B	STORE QUOTIENT
1636 2D	3531	DCR	L	DECREMENT DIGIT COUNT
1637 C2 D4 15	3532	JNZ	DIV4	
163A 21 06 17	3533	LXI	H,EXPD	
163D C1	3534	POP	B	
163E CD 9D 16	3535	CALL	STORO	
1641 C9	3536	RET		
1642	3537	* FETCH AND ALIGN ARGUMENTS FOR		
1642	3538	* ADDITION AND SUBTRACTION		
1642 1A	3539	EXPCK	LDAX	D
1643 96	3540	SUB	M	DIFFERENCE OF EXPS
1644 0E 00	3541	MVI	C,0	
1646 D2 4D 16	3542	JNC	EXPC1	
1649 0C	3543	INR	C	
164A EB	3544	XCHG		
164B 2F	3545	CMA		
164C 3C	3546	INR	A	
164D 47	3547	EXPCK	MOV	B,A
164E 1A	3548	LDAX	D	
164F 32 29 17	3549	STA	EXP	
1652 78	3550	MOV	A,B	
1653 FE 06	3551	CPI	DIGIT+DIGIT	
1655 DA 5A 16	3552	JC	EXPC2	
1658 3E 06	3553	MVI	A,DIGIT+DIGIT	
165A 07	3554	EXPCK	RLC	
165B 07	3555	RLC		
165C 47	3556	MOV	B,A	
165D E6 04	3557	ANI	4	
165F 32 2A 17	3558	STA	RCTRL	SET ROUNDING CONTROL
1662 C5	3559	PUSH	B	
1663 D5	3560	PUSH	D	
1664 CD 84 16	3561	CALL	LOAD	LOAD SMALLER VALUE
1667 3E 28	3562	MVI	A,8*DIGIT+16	

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

1669 90	3563	SUB	B	
166A FE 28	3564	CPI	8*DIGIT+16	
166C CA 7E 46	3565	JZ	EXPC3	
166F E6 F8	3566	ANI	OF8H	
1671 1F	3567	RAR		
1672 1F	3568	RAR		
1673 1F	3569	RAR		
1674 83	3570	ADD	E	
1675 5F	3574	MOV	E,A	
1676 7A	3572	MOV	A,D	
1677 CE 00	3573	ACI	O	
1679 57	3574	MOV	D,A	
167A 1A	3575	LDAX	D	GET ROUNDING DIGIT
167B 32 2B 47	3576	STA	RDIGI	SAVE
167E CD A8 46	3577 EXPC3	CALL	RIGHT	ALIGN VALUES
1681 D1	3578	POP	D	
1682 C1	3579	POP	B	
1683 C9	3580	RET		
1684	3581	*LOAD ARGUMENT INTO BUFFER		
1684 11 28 17	3582 LOAD	LXI	D,SIGN	
1687 0E 04	3583	MVI	C,DIGIT+1	
1689 2B	3584	DCX	H	
168A 7E	3585 LOAD1	MOV	A,M	
168B 12	3586	STAX	D	
168C 2B	3587	DCX	H	
168D 1B	3588	DCX	D	
168E 0D	3589	DCR	C	
168F C2 8A 46	3590	JNZ	LOAD1	
1692 AF	3591	XRA	A	
1693 12	3592	STAX	D	
1694 1B	3593	DCX	D	
1695 12	3594	STAX	D	
1696 32 2B 17	3595	STA	RDIGI	ZERO ROUNDING DIGIT
1699 C9	3596	RET		
169A	3597	* STORE RESULTS IN MEMORY		
169A 21 29 17	3598 STORE	LXI	H,EXP	
169D 1E 05	3599 STOR0	MVI	E,DIGIT+2	
169F 7E	3600 STOR1	MOV	A,M	
16A0 02	3601	STAX	B	
16A1 0B	3602	DCX	B	
16A2 2B	3603	DCX	H	
16A3 1D	3604	DCX	E	
16A4 C2 9F 16	3605	JNZ	STOR1	
16A7 C9	3606	RET		
16A8	3607	* SHIFT RIGHT NUMBER OF DIGITS		
16A8	3608	* IN B/4		
16A8 0F 04	3609 RIGHT	MVI	C,DIGIT+1	
16AA 21 24 17	3610 RIGHT1	LXI	H,BUF-1	
16AC 7E	3611	MOV	A,B	
16AE 06 08	3612	SUI	A	CHECK IF BYTE CAN BE SHIFTED
16B0 D2 C1 16	3613	JNC	RIGHT3	
16B3 05	3614	JCF	B	
16B4 F8	3615	RM		
16B5 B7	3616	ORA	A	
16B6 7E	3617 RIGHT2	MOV	A,M	
16B7 1F	3618	RAR		

16B8 77	3619	MOV	M,A
16B9 23	3620	INX	H
16BA 0D	3621	DCR	C
16BB C2 B6 16	3622	JNZ	RIGH2
16BE C3 A8 16	3623	JMP	RIGHT
16C1	3624	* SHIFT RIGHT ONE BYTE	
16C1 47	3625	RIGH3	MOV B,A
16C2 AF	3626	XRA	A
16C3 56	3627	RIGH4	MOV D,M
16C4 77	3628	MOV	M,A
16C5 7A	3629	MOV	A,D
16C6 23	3630	INX	H
16C7 0D	3631	DCR	C
16C8 C2 C3 16	3632	JNZ	RIGH4
16CB C3 A8 16	3633	JMP	RIGHT
16CE	3634	* SHIFT LEFT NUMBER OF DIGITS	
16CE	3635	* IN B/4	
16CE 0E 04	3636	LEFT	MVI C,DIGIT+1
16D0 21 27 17	3637	LXI	H,SIGN-1
16D3 78	3638	LEF1	MOV A,B
16D4 D6 08	3639	SUI	8
16D6 D2 E7 16	3640	JNC	LEF3
16D9 05	3641	DCR	B
16DA F8	3642	RM	
16DB B7	3643	ORA	A
16DC 7E	3644	LEF2	MOV A,M
16DD 17	3645	RAL	
16DE 77	3646	MOV	M,A
16DF 2B	3647	DCX	H
16E0 0D	3648	DCR	C
16E1 C2 DC 16	3649	JNZ	LEF2
16E4 C3 CE 16	3650	JMP	LEFT
16E7	3651	* SHIFT LEFT ONE BYTE	
16E7 47	3652	LEF3	MOV B,A
16E8 AF	3653	XRA	A
16E9 56	3654	LEF4	MOV D,M
16EA 77	3655	MOV	M,A
16EB 7A	3656	MOV	A,D
16EC 2B	3657	DCX	H
16ED 0D	3658	DCR	C
16EE C2 E9 16	3659	JNZ	LEF4
16F1 C3 CE 16	3660	JMP	LEFT
16F4	3661	* SET FLAGS FOR OVERFLOW,UNDERFLOW,	
16F4	3662	* AND DIVIDE BY ZERO	
16F4 01 50 46	3663	OVER	LXI B,'FP'
16F7 C3 C5 00	3664	JMP	ERROR
16FA 3E FF	3665	UNDER	MVI A,-1
16FC 32 23 17	3666	STA	ERRI
16FF 33	3667	INSP	INX SP
1700 33	3668	INX	SP
1701 C9	3669	RET	
1702	3670	DIVZ	EQU OVER
1702	3671	*	
1702	3672	* FLOATING POINT RAM	
1702	3673	*	
1702	3674	HOLD1	DS DIGIT+1

1706	3675	HOLD2	DS	DIGIT+1
170A	3676	HOLD3	DS	DIGIT+1
170E	3677	HOLD4	DS	DIGIT+1
1712	3678	HOLD5	DS	DIGIT+1
1716	3679	HOLD6	DS	DIGIT+1
171A	3680	HOLD7	DS	DIGIT+1
171E	3681	HOLD8	DS	DIGIT+1
1722	3682		DS	1
1723	3683	ERRI	DS	1 ERROR FLAG
1724	3684		DS	1
1725	3685	BUF	DS	DIGIT WORKING BUFFER
1728	3686	SIGN	DS	1 SIGN BIT
1729	3687	EXP	DS	1 EXPONENT
172A	3688	RCTRL	DS	1 ROUNDING CONTROL FLAG 1=MSD
172B	3689	RDIGI	DS	1 ROUNDING DIGIT
172C	3690	SIGND	EQU	HOLD1+DIGIT
172C	3691	EXPD	EQU	HOLD1+DIGIT+1
172C	3692	*		
172C	3693	*	SYSTEM RAM	
172C	3694	*		
172C	3695	EROM	DS	0
172C	3696		DS	100
1790	3697	CMNDSP	DS	0
1790	3698	PHEAD	DS	1
1791	3699	RELTYP	DS	1
1792	3700	NULLCT	DS	1
1793	3701	ARGF	DS	1
1794	3702	DIRF	DS	1
1795	3703	TXA	DS	2
1797	3704	CSTKSZ	EQU	100
1797	3705	ASTKSZ	EQU	FPSIZ*LINLEN/2
1797	3706	CSTKL	DS	CSTKSZ
17FB	3707	ASTKL	DS	ASTKSZ
18B1	3708	RTXA	DS	2
18B3	3709	STA	DS	2
18B5	3710	CSTKA	DS	2
18B7	3711	SINK	DS	FPSIZ-1
18BB	3712	FPSINK	DS	1
18BC	3713		DS	FPSIZ-1
18C0	3714	FTEMP	DS	1
18C1	3715		DS	FPSIZ-1
18C5	3716	FTEM4	DS	1
18C6	3717		DS	FPSIZ-1
18CA	3718	FTEM2	DS	1
18CB	3719		DS	FPSIZ-1
18CF	3720	FRAND	DS	1
18D0	3721	IBCNT	DS	1
18D1	3722	IBLN	DS	2
18D3	3723	IBUF	DS	LINLEN
191C	3724	ASTKA	DS	2
191E	3725	MATA	DS	2
1920	3726	ADDS	DS	2
1922	3727	ADDT	DS	2
1924	3728	BCADD	DS	2
1926	3729	OPST	DS	1
1927	3730	OPSTR	DS	1

PROCESSOR TECHNOLOGY BASIC 5
 ** COPYRIGHT 1976 **

PROCESSOR TECHNOLOGY CORP.
 6200 HOLLIS STREET
 EMERYVILLE, CALIF. 94608

1928	3731	ECNT	DS	1	
1929	3732	FSIGN	DS	1	
192A	3733	BC	DS		DIGIT+2
192F	3734	ABUF	DS		DIGIT*2+2
1937	3735	XSIGN	DS	1	
1938	3736	EXPO	DS	1	
1939	3737	FES	DS	1	
193A	3738	INFES	DS	1	
193B	3739	MAXL	DS	2	
193D	3740	INSA	DS	2	
193F	3741	*			
193F	3742	*	VDM GLOBAL		
193F	3743	*			
193F	3744	BRKCHR	DS	1	BREAK CHR STORAGE
1940	3745	CLN	DS	1	CURRENT SCREEN LINE NUMBER
1941	3746	CCP	DS	1	CURRENT CURSOR POSITION
1942	3747	CURF	DS	1	CURSOR DISPLAY SWITCH
1943	3748	BOSL	DS	1	BEGINNING OF SCREEN LINE
1944	3749	BOTL	DS	1	BOTTOM OF SCREEN LINE
1945	3750	*			
1945	3751	VDMDEV	EQU	0C8H	VDM PORT ADDRESS FOR SCROLL CONTROL
1945	3752	VDMBASE	EQU	0CC00H	VDM SCREEN MEMORY ADDRESS
1945	3753	VDMPAGE	EQU	VDMBASE/256	
1945	3754	SPEED	DS	1	
1946	3755	*			
1946	3756	*	SPECIAL INTERFACE GLOBAL		
1946	3757	*			
1946	3758	CALST	DS	6	
194C	3759	CALLA	DS	2	
194E	3760	EOFA	DS	2	END OF FILE ADDRESS
1950	3761	BOFA	DS	2	START OF FILE ADDRESS
1952	3762	MEMTOP	DS	2	STORAGE FOR LAST ASSIGNED MEMORY LOCATION
1954	3763	*			
1954	3764	*			