

PASCAL COMPUTER LANGUAGE INFORMATION

0001:	(* \$U+ COMPILE TO 72 COLUMNS ONLY.	*)	PASCODE	2	
0002:	(* \$A+ USE ASCII CHARACTER SET.	*)	PASCODE	3	
0003:	(* \$T- COMPILE WITHOUT RUN-TIME TESTS.	*)	PASCODE	4	
0004:	(* \$P- COMPILE WITH MINIMAL PMD INFO.	*)	PASCODE	5	
0005:	(* \$R- DO NOT REDUCE FIELD LENGTH.	*)	PASCODE	6	
0006:	(* \$E+ MEANINGFUL ENTRY-POINT NAMES.	*)	PASCODE	7	
0007:			PASCODE	8	
0008:			PASCODE	9	
0009:	PROGRAM PASCODE(LGO,OUTPUT);		PASCODE	10	
0010:			PASCODE	11	
0011:			PASCODE	12	
0012:			PASCODE	13	
0013:			PASCODE	14	
0014:	(* - - - - -	*)	PASCODE	15	
0015:	(*	*)	PASCODE	16	
0016:	(* PASCODE - LIST RELOCATABLE BINARY CODE IN SYMBOLIC FORM.	*)	PASCODE	17	
0017:	(*	*)	PASCODE	18	
0018:	(* ORIGINAL PROGRAM (DECODE).	N.WIRTH, MAY 74	*)	PASCODE	19
0019:	(* LIST EXTERNALS AND OTHER NICE STUFF.	J.STRAIT, OCT 76	*)	PASCODE	20
0020:	(* HANDLE NON-PASCAL BINARIES.	J.STRAIT, OCT 77	*)	PASCODE	21
0021:	(* IMPROVE TITLE INFO, UTILIZE VALUE.	J.STRAIT, DEC 77	*)	PASCODE	22
0022:	(* ADD OPTIONAL OP-CODE COUNT.	J.STRAIT, APR 78	*)	PASCODE	23
0023:	(* CLEAN UP / ADD COMMENTS.	J.STRAIT, MAY 78	*)	PASCODE	24
0024:	(* BETTER REPORTING OF NO-OP STATISTICS.	J.DRUMMOND, SEP 78	*)	PASCODE	25
0025:	(* FINAL CLEANUP AND CORRECTIONS.	J.STRAIT, NOV 78	*)	PASCODE	26
0026:	(*	*)	PASCODE	27	
0027:	(* PASCODE LISTS BINARY CODE PRODUCED BY THE PASCAL COMPILER	*)	PASCODE	28	
0028:	(* IN PSEUDO-COMPASS. TO CONSERVE SPACE, ONE WORD OF INSTRUCTIONS	*)	PASCODE	29	
0029:	(* IS PRINTED PER LINE IN COLUMNS WHICH CORRESPOND TO THE FOUR	*)	PASCODE	30	
0030:	(* PARCELS IN CENTRAL MEMORY WORDS. IF PASCODE IS PRESENTED WITH A	*)	PASCODE	31	
0031:	(* PASCAL BINARY WHICH WAS NOT COMPILED WITH POST MORTEM DUMP INFO	*)	PASCODE	32	
0032:	(* ENTIRELY SUPPRESSED, IT IS ABLE TO DIFFERENTIATE BETWEEN CODE,	*)	PASCODE	33	
0033:	(* CONSTANTS AND POST MORTEM DUMP INFO. IF IT IS PRESENTED WITH A	*)	PASCODE	34	
0034:	(* NON-PASCAL BINARY, OR ONE COMPILED WITH PMD INFO SUPPRESSED (P0),	*)	PASCODE	35	
0035:	(* PASCODE WILL PRINT EVERY WORD IN BOTH OCTAL AND PSEUDO-COMPASS.	*)	PASCODE	36	
0036:	(*	*)	PASCODE	37	
0037:	(* PASCODE PRINTS CORRECT PSEUDO-COMPASS FOR PASCAL BINARIES,	*)	PASCODE	38	
0038:	(* AND DOES FAIRLY WELL FOR NON-PASCAL BINARIES. PASCODE DOES NOT	*)	PASCODE	39	
0039:	(* HANDLE ALL OF THE DEFINED LOADER TABLES, ONLY THOSE THAT THE	*)	PASCODE	40	
0040:	(* PASCAL COMPILER PRODUCES. IT DOES NOT HANDLE MORE THAN ONE	*)	PASCODE	41	
0041:	(* EXTERNAL REFERENCE APPLIED TO A SINGLE ADDRESS FIELD, NOR DOES	*)	PASCODE	42	
0042:	(* IT HANDLE AN EXTERNAL REFERENCE AND PROGRAM RELOCATION APPLIED TO	*)	PASCODE	43	
0043:	(* THE SAME ADDRESS FIELD. COMMON BLOCKS ARE NOT HANDLED. ONLY	*)	PASCODE	44	
0044:	(* ONE ENTRY POINT IS RECOGNIZED IN EACH RELOCATABLE RECORD. EACH	*)	PASCODE	45	
0045:	(* PROCEDURE IS ASSUMED TO BE TERMINATED BY AN EOR MARK.	*)	PASCODE	46	
0046:	(*	*)	PASCODE	47	
0047:	(* IN PARTICULAR, THE ONLY LOADER TABLES THAT ARE PROCESSED ARE:	*)	PASCODE	48	
0048:	(*	*)	PASCODE	49	
0049:	(* 3400 - PIDL - PROGRAM IDENTIFICATION AND LENGTH.	*)	PASCODE	50	
0050:	(* 3600 - ENTR - ENTRY POINT DEFINITIONS.	*)	PASCODE	51	
0051:	(* 4000 - TEXT - RELOCATABLE TEXT.	*)	PASCODE	52	
0052:	(* 4300 - REPL - REPLICATION OF TEXT.	*)	PASCODE	53	
0053:	(* 4400 - LINK - EXTERNAL REFERENCE LINKAGE.	*)	PASCODE	54	
0054:	(* 7700 - PRFX - PREFIX.	*)	PASCODE	55	
0055:	(*	*)	PASCODE	56	
0056:	(* ALL OTHER TABLES ARE IGNORED.	*)	PASCODE	57	
0057:	(*	*)	PASCODE	58	
0058:	(* THE PROGRAM HEADING OF PASCODE IS:	*)	PASCODE	59	
0059:	(* PROGRAM PASCODE(LGO,OUTPUT)	*)	PASCODE	60	
0060:	(* AND SO TO DECODE A BINARY ON FILE LGO USE THE CONTROL STATEMENT:	*)	PASCODE	61	
0061:	(* PASCODE.	*)	PASCODE	62	
0062:	(* AND TO DECODE A BINARY ON FILE ABC USE THE CONTROL STATEMENT:	*)	PASCODE	63	
0063:	(* PASCODE(ABC)	*)	PASCODE	64	
0064:	(*	*)	PASCODE	65	
0065:	(* - - - - -	*)	PASCODE	66	

PASCAL COMPUTER LANGUAGE INFORMATION

0066:		PASCODE	67
0067:		PASCODE	68
0068:		PASCODE	69
0069:		PASCODE	70
0070:	CONST TBLLEN = 64;	(* TABLE LENGTH FOR SAVING BINARY *)	PASCODE 71
0071:	EXTMAX = 300;	(* MAX EXTERNAL REFS IN ONE PROCEDURE *)	PASCODE 72
0072:	SLEWSIZE = 4;	(* BLANKS LINES BETWEEN PROCEDURES *)	PASCODE 73
0073:	MINPAGE = 7;	(* MIN LINECOUNT AT BOTTOM OF PAGE *)	PASCODE 74
0074:	PAGESIZE = 60;	(* LINES PRINTED PER PAGE *)	PASCODE 75
0075:		PASCODE	76
0076:	BYTESHIFT = 10000B;	PASCODE	77
0077:	PARCELSHIFT = 100000B;	PASCODE	78
0078:	HALFSHIFT = 10000000000B;	PASCODE	79
0079:		PASCODE	80
0080:	PIDL = 34B;	(* PROGRAM IDENTIFICATION AND LENGTH *)	PASCODE 81
0081:	ENTR = 36B;	(* ENTRY POINT DEFINITIONS *)	PASCODE 82
0082:	XTEXT = 37B;	(* EXTENDED RELOCATABLE TEXT *)	PASCODE 83
0083:	TEXT = 40B;	(* RELOCATABLE TEXT *)	PASCODE 84
0084:	XFILL = 41B;	(* EXTENDED RELOCATION FILL *)	PASCODE 85
0085:	FILL = 42B;	(* RELOCATION FILL *)	PASCODE 86
0086:	REPL = 43B;	(* REPLICATION OF TEXT *)	PASCODE 87
0087:	LINK = 44B;	(* EXTERNAL REFERENCE LINKAGE *)	PASCODE 88
0088:	XLINK = 45B;	(* EXTENDED EXTERNAL REFERENCE LINKAGE *)	PASCODE 89
0089:	XFER = 46B;	(* TRANSFER POINT *)	PASCODE 90
0090:	XREPL = 47B;	(* EXTENDED REPLICATION OF TEXT *)	PASCODE 91
0091:	LDSET = 70B;	(* OBJECT DIRECTIVE *)	PASCODE 92
0092:	PRFX = 77B;	(* PREFIX *)	PASCODE 93
0093:		PASCODE	94
0094:	TYPE OPCODE = 0..77B;	(* 6 BIT OP CODE *)	PASCODE 95
0095:	REGISTER = 0..7B;	(* 3 BIT REGISTER *)	PASCODE 96
0096:	BYTEFIELD = 0..7777B;	(* 12 BIT BYTE *)	PASCODE 97
0097:	PARCELFIELD = 0..77777B;	(* 15 BIT PARCEL *)	PASCODE 98
0098:	ADDRESSFIELD = 0..777777B;	(* 18 BIT ADDRESS *)	PASCODE 99
0099:	SIGNEDADDRESS = -377777B..377777B;	(* 18 BIT SIGNED ADDRESS *)	PASCODE 100
0100:	HALFFIELD = 0..7777777777B;	(* 30 BIT HALFWORD *)	PASCODE 101
0101:		PASCODE	102
0102:	WORD = (* WAYS OF LOOKING AT A MEMORY WORD *)	PASCODE	103
0103:	PACKED RECORD CASE INTEGER OF	PASCODE	104
0104:	1 : (* 60 BIT INTEGER *)	PASCODE	105
0105:	(N : INTEGER);	PASCODE	106
0106:	2 : (* 60 BIT REAL *)	PASCODE	107
0107:	(R : REAL);	PASCODE	108
0108:	3 : (* 10 CHARACTERS *)	PASCODE	109
0109:	(NAME : ALFA);	PASCODE	110
0110:	4 : (* 60 BITS *)	PASCODE	111
0111:	(BIT : PACKED ARRAY[-59..0] OF BOOLEAN);	PASCODE	112
0112:	5 : (* 20 3 BIT DIGITS *)	PASCODE	113
0113:	(DIGIT : PACKED ARRAY[1..20] OF 0..7);	PASCODE	114
0114:	6 : (* 5 12 BIT BYTES *)	PASCODE	115
0115:	(BYTE : PACKED ARRAY[0..4] OF BYTEFIELD);	PASCODE	116
0116:	7 : (* 4 15 BIT PARCELS *)	PASCODE	117
0117:	(PARCEL : PACKED ARRAY[0..3] OF PARCELFIELD);	PASCODE	118
0118:	8 : (* 3 18 BIT ADDRESSES *)	PASCODE	119
0119:	(ADDRESS : PACKED RECORD	PASCODE	120
0120:	PAD : 0..77B;	PASCODE	121
0121:	UPPER : ADDRESSFIELD;	PASCODE	122
0122:	MIDDLE : ADDRESSFIELD;	PASCODE	123
0123:	LOWER : ADDRESSFIELD	PASCODE	124
0124:	END);	PASCODE	125
0125:	9 : (* 2 30 BIT HALFWORDS *)	PASCODE	126
0126:	(HALFWORD : PACKED ARRAY[0..1] OF HALFFIELD);	PASCODE	127
0127:	10 : (* 15 BIT INSTRUCTION *)	PASCODE	128
0128:	(QUARTER : PACKED RECORD	PASCODE	129
0129:	OP : OPCODE;	PASCODE	130
0130:	I : REGISTER;	PASCODE	131

PASCAL COMPUTER LANGUAGE INFORMATION

```

0131:          J      : REGISTER;                      PASCODE  132
0132:          K      : REGISTER                      PASCODE  133
0133:          END);                                  PASCODE  134
0134:    11 : (* 30 BIT INSTRUCTION *)                 PASCODE  135
0135:          (HALF : PACKED RECORD                   PASCODE  136
0136:            OP   : OPCODE;                        PASCODE  137
0137:            I    : REGISTER;                      PASCODE  138
0138:            J    : REGISTER;                      PASCODE  139
0139:            K    : SIGNEDADDRESS                 PASCODE  140
0140:          END);                                  PASCODE  141
0141:    12 : (* 7 CHARACTERS *)                       PASCODE  142
0142:          (SHORTNAME : PACKED ARRAY[1..7] OF CHAR); PASCODE  143
0143:    13 : (* BINARY TABLE HEADER WORD *)          PASCODE  144
0144:          (HEADER : PACKED RECORD                 PASCODE  145
0145:            TAG   : BYTEFIELD;                    PASCODE  146
0146:            WCNT  : BYTEFIELD                     PASCODE  147
0147:          END);                                  PASCODE  148
0148:    14 : (* RELOCATION WORD *)                     PASCODE  149
0149:          (REL : PACKED ARRAY[1..15] OF 1..17B); PASCODE  150
0150:    15 : (* EXTERNAL REFERENCE TRAILER BYTE *)   PASCODE  151
0151:          (TRAILER : PACKED RECORD               PASCODE  152
0152:            T : BOOLEAN;                          PASCODE  153
0153:            P : 0..2;                              PASCODE  154
0154:            R : 0..777B;                           PASCODE  155
0155:            A : ADDRESSFIELD                       PASCODE  156
0156:          END);                                  PASCODE  157
0157:    16 : (* REPL TABLE WORD 1 *)                PASCODE  158
0158:          (REPL1 : PACKED RECORD                 PASCODE  159
0159:            PAD   : 0..77777B;                    PASCODE  160
0160:            INC   : 0..777777B;                  PASCODE  161
0161:            SR    : 0..777B;                     PASCODE  162
0162:            SADDR : 0..777777B                   PASCODE  163
0163:          END);                                  PASCODE  164
0164:    17 : (* REPL TABLE WORD 2 *)                PASCODE  165
0165:          (REPL2 : PACKED RECORD                 PASCODE  166
0166:            COUNT : 0..777777B;                  PASCODE  167
0167:            BSIZE : 0..77777B;                   PASCODE  168
0168:            DR    : 0..777B;                      PASCODE  169
0169:            DADDR : 0..777777B                   PASCODE  170
0170:          END);                                  PASCODE  171
0171:    18 : (* PROCEDURE DESCRIPTOR WORD *)         PASCODE  172
0172:          (DESCRIPTOR : PACKED RECORD           PASCODE  173
0173:            P      : BOOLEAN;                     PASCODE  174
0174:            S      : BOOLEAN;                     PASCODE  175
0175:            F      : BOOLEAN;                     PASCODE  176
0176:            PAD    : 0..77B;                       PASCODE  177
0177:            COUNT  : 0..77B;                       PASCODE  178
0178:            ENTRY  : PARSEFIELD;                  PASCODE  179
0179:            CONSTS : PARSEFIELD;                  PASCODE  180
0180:            PMDINF : PARSEFIELD                   PASCODE  181
0181:          END);                                  PASCODE  182
0182:    19 : (* POST MORTEM DUMP INFO *)             PASCODE  183
0183:          (INFO : PACKED RECORD                 PASCODE  184
0184:            PAD   : 0..77777777B;                PASCODE  185
0185:            VTYPE : ADDRESSFIELD;                PASCODE  186
0186:            VADDR : ADDRESSFIELD                 PASCODE  187
0187:          END);                                  PASCODE  188
0188:          END;                                  PASCODE  189
0189:          PASCODE  190
0190:          (* FOR REMEMBERING TEXT BLOCKS BETWEEN PASS1 AND PASS2 *) PASCODE  191
0191:          PTEXTBLOCK = ^TEXTBLOCK;              PASCODE  192
0192:          TEXTBLOCK = RECORD                     PASCODE  193
0193:            T      : ARRAY[1..TBLEN] OF WORD;    PASCODE  194
0194:            NXTTB  : PTEXTBLOCK                  PASCODE  195
0195:          END;                                  PASCODE  196

```

PASCAL COMPUTER LANGUAGE INFORMATION

0196:		PASCODE	197
0197:	(* FOR THE INDEX PRINTED AT THE END OF THE TITLES *)	PASCODE	198
0198:	PINDEXREC = ^INDEXREC;	PASCODE	199
0199:	INDEXREC = RECORD	PASCODE	200
0200:	NAME,ZNAME : ALFA;	PASCODE	201
0201:	ENTRYNAME : ALFA;	PASCODE	202
0202:	PAGE : INTEGER;	PASCODE	203
0203:	LENGTH : INTEGER;	PASCODE	204
0204:	NEXT : PINDEXREC	PASCODE	205
0205:	END;	PASCODE	206
0206:		PASCODE	207
0207:	VAR (* FILE TO DECODE *)	PASCODE	208
0208:	LGO : SEGMENTED FILE OF WORD;	PASCODE	209
0209:		PASCODE	210
0210:	(* STRUCTURED CONSTANTS *)	PASCODE	211
0211:	CODE : ARRAY[OPCODE] OF ALFA;	PASCODE	212
0212:	(* TEXT OF INSTRUCTIONS *)	PASCODE	213
0213:	TEST : ARRAY[0..7] OF PACKED ARRAY[1..2] OF CHAR;	PASCODE	214
0214:	(* TEXT OF TEST-X INSTRUCTIONS *)	PASCODE	215
0215:	TYP : ARRAY[1..8] OF ALFA;	PASCODE	216
0216:	(* TEXT OF PMD TYPE NAMES *)	PASCODE	217
0217:	TABLES : SET OF 0..58; (* REPROCESSED SET OF TABLE NUMBERS *)	PASCODE	218
0218:		PASCODE	219
0219:	(* LISTING *)	PASCODE	220
0220:	TODAY,NOW : ALFA;	PASCODE	221
0221:	TITLEMESSAGE : PACKED ARRAY[1..30] OF CHAR;	PASCODE	222
0222:	LTITLEMESSAGE : INTEGER;	PASCODE	223
0223:	LPROCNAME : INTEGER;	PASCODE	224
0224:	CREATION : ARRAY[1..5] OF ALFA;	PASCODE	225
0225:	LINECOUNT,PAGECOUNT : INTEGER;	PASCODE	226
0226:		PASCODE	227
0227:	(* INDEX *)	PASCODE	228
0228:	INDEXBASE,INDEXTHIS : PINDEXREC;	PASCODE	229
0229:	INDEXCOUNT : INTEGER;	PASCODE	230
0230:		PASCODE	231
0231:	(* REMEMBRANCE OF THE BINARY *)	PASCODE	232
0232:	FSTTB,LSTTB,TB : PTEXTBLOCK;	PASCODE	233
0233:	TBI,TBL : INTEGER;	PASCODE	234
0234:	EPTNAME,BLOCKNAME,PROCNAME : ALFA;	PASCODE	235
0235:	EPTADDR : ADDRESSFIELD;	PASCODE	236
0236:	FOREIGN : BOOLEAN;	PASCODE	237
0237:	EXTC : INTEGER;	PASCODE	238
0238:	EXT : ARRAY[1..EXTMAX] OF RECORD	PASCODE	239
0239:	NAME : ALFA;	PASCODE	240
0240:	ADDR : ADDRESSFIELD;	PASCODE	241
0241:	POS : 0..2	PASCODE	242
0242:	END;	PASCODE	243
0243:		PASCODE	244
0244:	(* STATISTICS *)	PASCODE	245
0245:	CODECOUNT : ARRAY[OPCODE] OF INTEGER;	PASCODE	246
0246:	TESTCOUNT : ARRAY[0..7] OF INTEGER;	PASCODE	247
0247:	NOOPS : ARRAY[BOOLEAN] OF INTEGER;	PASCODE	248
0248:	BIGNOOPS : ARRAY[BOOLEAN] OF INTEGER;	PASCODE	249
0249:	TOTALCOUNT : INTEGER;	PASCODE	250
0250:	PROCLENGTH : INTEGER;	PASCODE	251
0251:		PASCODE	252
0252:		PASCODE	253
0253:	VALUE	PASCODE	254
0254:		PASCODE	255
0255:	(* THE OPCODE TABLE CONTROLS DECOMPOSITION OF INSTRUCTIONS *)	PASCODE	256
0256:	(* BY USING THE REPLACEMENTS DEFINED HERE. NOTE THAT SOME OF *)	PASCODE	257
0257:	(* THE REPLACEMENTS ARE USED SOLELY TO MAKE THE COLUMNIZING *)	PASCODE	258
0258:	(* COME OUT RIGHT. *)	PASCODE	259
0259:	(* *)	PASCODE	260
0260:	(* 0 - REGISTER I. *)	PASCODE	261

PASCAL COMPUTER LANGUAGE INFORMATION

0261:	(*	1 - REGISTER J.	*)	PASCODE	262			
0262:	(*	2 - REGISTER K.	*)	PASCODE	263			
0263:	(*	3 - ADDRESS K.	*)	PASCODE	264			
0264:	(*	4 - PLUS OR MINUS SIGN.	*)	PASCODE	265			
0265:	(*	5 - X-REGISTER TEST OP CODE.	*)	PASCODE	266			
0266:	(*	6 - (UNUSED).	*)	PASCODE	267			
0267:	(*	7 - JK (MASK OR SHIFT COUNT).	*)	PASCODE	268			
0268:	(*	8 - 3 BLANKS.	*)	PASCODE	269			
0269:	(*	9 - NULL.	*)	PASCODE	270			
0270:	(*	. - TRAILING BLANKS	*)	PASCODE	271			
0271:	(*		*)	PASCODE	272			
0272:	(*	ALL OTHERS ARE LITERAL CHARACTERS.	*)	PASCODE	273			
0273:				PASCODE	274			
0274:	CODE = ('PS	3...8','RJ	3...8','JP	B0438','59	X1,38',	PASCODE	275	
0275:		'EQ8B0,B1,3',	'NE8B0,B1,3',	'GE8B0,B1,3',	'LT8B0,B1,3',	PASCODE	276	
0276:		'BX0	X1...','BX0	X1*X2','BX0	X1+X2','BX0	X1-X2',	PASCODE	277
0277:		'BX0	-X2...','BX0	-X2*X1','BX0	-X2+X1','BX0	-X2-X1',	PASCODE	278
0278:		'LX0	7...9','AX0	7...9','LX0	B1,X2','AX0	B1,X2',	PASCODE	279
0279:		'NX0	B1,X2','ZX0	B1,X2','UX0	B1,X2','PX0	B1,X2',	PASCODE	280
0280:		'FX0	X1+X2','FX0	X1-X2','DX0	X1+X2','DX0	X1-X2',	PASCODE	281
0281:		'RX0	X1+X2','RX0	X1-X2','IX0	X1+X2','IX0	X1-X2',	PASCODE	282
0282:		'FX0	X1*X2','RX0	X1*X2','DX0	X1*X2','MX0	7...9',	PASCODE	283
0283:		'FX0	X1/X2','RX0	X1/X2','NO.....',	'CX0	X2...'	PASCODE	284
0284:		'SA0	A1438','SA0	B1438','SA0	X1438','SA0	X1+B2',	PASCODE	285
0285:		'SA0	A1+B2','SA0	A1-B2','SA0	B1+B2','SA0	B1-B2',	PASCODE	286
0286:		'SB0	A1438','SB0	B1438','SB0	X1438','SB0	X1+B2',	PASCODE	287
0287:		'SB0	A1+B2','SB0	A1-B2','SB0	B1+B2','SB0	B1-B2',	PASCODE	288
0288:		'SX0	A1438','SX0	B1438','SX0	X1438','SX0	X1+B2',	PASCODE	289
0289:		'SX0	A1+B2','SX0	A1-B2','SX0	B1+B2','SX0	B1-B2');	PASCODE	290
0290:							PASCODE	291
0291:	TEST = ('ZR',	'NZ','PL',	'NG','IR',	'OR','DF',	'ID');		PASCODE	292
0292:							PASCODE	293
0293:	TYP = ('INTEGER	','REAL	','CHAR	','BOOLEAN	','		PASCODE	294
0294:		'SCALAR	','ALFA	','POINTER	','POINTER	');	PASCODE	295
0295:							PASCODE	296
0296:	TITLEMESSAGE =	'DECODE OF					PASCODE	297
0297:	LTITLEMESSAGE =	10;					PASCODE	298
0298:	CODECOUNT =	(100B OF 0);					PASCODE	299
0299:	TESTCOUNT =	(10B OF 0);					PASCODE	300
0300:	NOOPS =	(0,0);					PASCODE	301
0301:	BIGNOOPS =	(0,0);					PASCODE	302
0302:	TOTALCOUNT =	0;					PASCODE	303
0303:							PASCODE	304
0304:							PASCODE	305
0305:							PASCODE	306
0306:							PASCODE	307
0307:	(* INDEX MAINTENANCE ROUTINES:					*)	PASCODE	308
0308:	(* SPACEFILL - SPACE FILL NAME.					*)	PASCODE	309
0309:	(* ZEROFILL - ZERO FILL NAME.					*)	PASCODE	310
0310:	(* NEWNAME - ADD NEW NAME TO THE INDEX.					*)	PASCODE	311
0311:							PASCODE	312
0312:							PASCODE	313
0313:	PROCEDURE SPACEFILL(VAR NAME: ALFA);						PASCODE	314
0314:	VAR I : INTEGER;						PASCODE	315
0315:	BEGIN (* SPACEFILL *)						PASCODE	316
0316:	FOR I := 1 TO 10 DO						PASCODE	317
0317:	IF NAME[I] = CHR(0)						PASCODE	318
0318:	THEN NAME[I] := ' '						PASCODE	319
0319:	END (* SPACEFILL *);						PASCODE	320
0320:							PASCODE	321
0321:							PASCODE	322
0322:	PROCEDURE ZEROFILL(VAR NAME : ALFA);						PASCODE	323
0323:	VAR I : INTEGER;						PASCODE	324
0324:	BEGIN (* ZEROFILL *)						PASCODE	325
0325:	FOR I := 1 TO 10 DO						PASCODE	326

```

0326:      IF NAME[I] = ' '                                PASCODE 327
0327:          THEN NAME[I] := CHR(0)                      PASCODE 328
0328:      END (* ZEROFILL *);                              PASCODE 329
0329:                                                    PASCODE 330
0330:                                                    PASCODE 331
0331:      PROCEDURE NEWNAME;                                PASCODE 332
0332:      VAR ZFNAME : ALFA;                                PASCODE 333
0333:          DONE : BOOLEAN;                               PASCODE 334
0334:          SCAN : PINDEXREC;                             PASCODE 335
0335:      BEGIN (* NEWNAME *)                               PASCODE 336
0336:          INDEXCOUNT := INDEXCOUNT + 1;              PASCODE 337
0337:          NEW(INDEXTHIS);                                PASCODE 338
0338:          SCAN := INDEXBASE;                             PASCODE 339
0339:          ZFNAME := PROCNAME;                            PASCODE 340
0340:          ZEROFILL(ZFNAME);                              PASCODE 341
0341:          REPEAT WITH SCAN^ DO                          PASCODE 342
0342:              BEGIN DONE := NEXT = NIL;                PASCODE 343
0343:                  IF NOT DONE THEN DONE := NEXT^.NAME > ZFNAME; PASCODE 344
0344:                  IF NOT DONE THEN SCAN := NEXT        PASCODE 345
0345:                  END                                  PASCODE 346
0346:          UNTIL DONE;                                    PASCODE 347
0347:          WITH INDEXTHIS^ DO                             PASCODE 348
0348:              BEGIN NAME := PROCNAME;                  PASCODE 349
0349:                  ZNAME := ZFNAME;                    PASCODE 350
0350:                  ENTRYNAME := EPTNAME;               PASCODE 351
0351:                  PAGE := PAGECOUNT;                  PASCODE 352
0352:                  NEXT := SCAN^.NEXT                   PASCODE 353
0353:              END;                                      PASCODE 354
0354:          SCAN^.NEXT := INDEXTHIS                       PASCODE 355
0355:          END (* NEWNAME *);                             PASCODE 356
0356:                                                    PASCODE 357
0357:                                                    PASCODE 358
0358:                                                    PASCODE 359
0359:                                                    PASCODE 360
0360:      (* PAGING ROUTINES:                                *) PASCODE 361
0361:      (* TITLE - PRINT PAGE TITLE.                       *) PASCODE 362
0362:      (* EJECT - EJECT TO NEW PAGE.                      *) PASCODE 363
0363:      (* BEGINLINE - CALLED BEFORE EACH LINE IS PRINTED. *) PASCODE 364
0364:      (* PROCTITLE - PRINT TITLE FOR NEW PROCEDURE.     *) PASCODE 365
0365:                                                    PASCODE 366
0366:                                                    PASCODE 367
0367:      PROCEDURE TITLE(P : BOOLEAN);                      PASCODE 368
0368:      BEGIN (* TITLE *)                                  PASCODE 369
0369:          IF P THEN WRITE('1') ELSE WRITE(' ');          PASCODE 370
0370:          WRITE(' ': 8, TITLEMESSAGE: LTITLEMESSAGE, PROCNAME: LPROCNAME, PASCODE 371
0371:              ', ' :40-LTITLEMESS-LPROCNAME, TODAY, NOW); PASCODE 372
0372:          IF P THEN WRITE(' PAGE ', PAGECOUNT: 1);    PASCODE 373
0373:          WRITELN                                       PASCODE 374
0374:          END (* TITLE *);                              PASCODE 375
0375:                                                    PASCODE 376
0376:                                                    PASCODE 377
0377:      PROCEDURE EJECT;                                   PASCODE 378
0378:      BEGIN (* EJECT *)                                  PASCODE 379
0379:          LINECOUNT := PAGESIZE                         PASCODE 380
0380:          END (* EJECT *);                              PASCODE 381
0381:                                                    PASCODE 382
0382:                                                    PASCODE 383
0383:      PROCEDURE BEGINLINE;                              PASCODE 384
0384:      BEGIN (* BEGINLINE *)                             PASCODE 385
0385:          IF LINECOUNT >= PAGESIZE                    PASCODE 386
0386:          THEN BEGIN PAGECOUNT := PAGECOUNT + 1;    PASCODE 387
0387:                  LINECOUNT := 1;                    PASCODE 388
0388:                  TITLE(TRUE);                          PASCODE 389
0389:                  WRITELN;                               PASCODE 390
0390:                  WRITELN                               PASCODE 391

```

```

0391:      END                                     PASCODE 392
0392:      ELSE LINECOUNT := LINECOUNT + 1     PASCODE 393
0393:      END (* BEGINLINE *);                  PASCODE 394
0394:                                             PASCODE 395
0395:                                             PASCODE 396
0396:      PROCEDURE PROCTITLE;                    PASCODE 397
0397:      VAR I : INTEGER;                        PASCODE 398
0398:      BEGIN (* PROCTITLE *)                  PASCODE 399
0399:      LPROCNAME := 1;                         PASCODE 400
0400:      WHILE (PROCNAME[LPROCNAME] <> ' ') AND (LPROCNAME < 10) DO PASCODE 401
0401:          LPROCNAME := LPROCNAME + 1;         PASCODE 402
0402:      IF PROCNAME[LPROCNAME] = ' ' THEN LPROCNAME := LPROCNAME - 1; PASCODE 403
0403:      IF LINECOUNT + SLEWSIZE + MINPAGE >= PAGESIZE THEN EJECT; PASCODE 404
0404:      BEGINLINE;                              PASCODE 405
0405:      IF LINECOUNT <> 1                      PASCODE 406
0406:          THEN BEGIN FOR I := 1 TO SLEWSIZE DO PASCODE 407
0407:              BEGIN WRITELN; BEGINLINE END; PASCODE 408
0408:              TITLE(FALSE);                  PASCODE 409
0409:              BEGINLINE;                    PASCODE 410
0410:              WRITELN; BEGINLINE;          PASCODE 411
0411:              WRITELN; BEGINLINE          PASCODE 412
0412:          END;                               PASCODE 413
0413:      NEWNAME;                               PASCODE 414
0414:      IF CREATION[1] <> ' '                   PASCODE 415
0415:          THEN BEGIN WRITE('          CREATED: ', CREATION[1]); PASCODE 416
0416:              FOR I := 2 TO 5 DO             PASCODE 417
0417:                  BEGIN IF (CREATION[I-1][10] <> ' ') AND PASCODE 418
0418:                      (CREATION[I ][ 1] <> ' ') PASCODE 419
0419:                      THEN WRITE(' '); PASCODE 420
0420:                      WRITE(CREATION[I]); PASCODE 421
0421:                  END;                       PASCODE 422
0422:              WRITELN;                       PASCODE 423
0423:              BEGINLINE                     PASCODE 424
0424:          END;                               PASCODE 425
0425:      WRITELN                                PASCODE 426
0426:      END (* PROCTITLE *);                   PASCODE 427
0427:                                             PASCODE 428
0428:                                             PASCODE 429
0429:                                             PASCODE 430
0430:                                             PASCODE 431
0431:      (* PASS1 - READ BINARY FILE.           *) PASCODE 432
0432:      (* DETERMINE ENTRYPOINT AND BLOCK NAMES. *) PASCODE 433
0433:      (* DETERMINE BLOCKLENGTH (IF GIVEN IN PIDL *) PASCODE 434
0434:      (* DETERMINE CREATION INFORMATION.     *) PASCODE 435
0435:      (* REMEMBER TEXT AND REPL TABLES.    *) PASCODE 436
0436:                                             PASCODE 437
0437:                                             PASCODE 438
0438:      PROCEDURE PASS1;                       PASCODE 439
0439:                                             PASCODE 440
0440:                                             PASCODE 441
0441:      PROCEDURE PROCESSTABLE; (* PROCESS ONE LOADER TABLE *) PASCODE 442
0442:      VAR W,EXTNAME,PROCESSOR : WORD; PASCODE 443
0443:          H,I,L : INTEGER; PASCODE 444
0444:                                             PASCODE 445
0445:                                             PASCODE 446
0446:      PROCEDURE ADDWORD(W : WORD); (* ADD A WORD TO REMEMBER *) PASCODE 447
0447:      BEGIN (* ADDWORD *) PASCODE 448
0448:      IF TBL = TBLEN PASCODE 449
0449:          THEN BEGIN NEW(TB); PASCODE 450
0450:              LSTTB^.NXTTB := TB; PASCODE 451
0451:              LSTTB := TB; PASCODE 452
0452:              TBL := 1 PASCODE 453
0453:          END PASCODE 454
0454:          ELSE TBL := TBL + 1; PASCODE 455
0455:          LSTTB^.T[TBL] := W PASCODE 456

```

0456:	END (* ADDWORD *);	PASCODE 457
0457:		PASCODE 458
0458:		PASCODE 459
0459:	PROCEDURE NEXTHALF; (* GET NEXT HALF WORD *)	PASCODE 460
0460:	BEGIN (* NEXTHALF *)	PASCODE 461
0461:	W.N := W.N * HALFSHIFT;	PASCODE 462
0462:	IF H = 0	PASCODE 463
0463:	THEN IF L > 0	PASCODE 464
0464:	THEN BEGIN GET(LGO); L := L - 1 END;	PASCODE 465
0465:	IF L > 0	PASCODE 466
0466:	THEN W.HALFWORD[1] := LGO^.HALFWORD[H]	PASCODE 467
0467:	ELSE W.HALFWORD[1] := 0;	PASCODE 468
0468:	H := 1 - H	PASCODE 469
0469:	END (* NEXTHALF *);	PASCODE 470
0470:		PASCODE 471
0471:		PASCODE 472
0472:	BEGIN (* PROCESSTABLE *)	PASCODE 473
0473:	W := LGO^;	PASCODE 474
0474:	L := W.HEADER.WCNT;	PASCODE 475
0475:	GET(LGO);	PASCODE 476
0476:	IF (W.HEADER.TAG MOD 100B = 0) AND	PASCODE 477
0477:	(W.HEADER.TAG DIV 100B - 30B IN TABLES)	PASCODE 478
0478:	THEN CASE W.HEADER.TAG DIV 100B OF	PASCODE 479
0479:	PIDL : BEGIN PROCLENGTH := LGO^.ADDRESS.LOWER;	PASCODE 480
0480:	FOR I := 1 TO L DO GET(LGO)	PASCODE 481
0481:	END;	PASCODE 482
0482:	ENTR : BEGIN EPTNAME := LGO^.NAME;	PASCODE 483
0483:	GET(LGO);	PASCODE 484
0484:	SPACEFILL(EPTNAME);	PASCODE 485
0485:	EPTADDR := LGO^.ADDRESS.LOWER;	PASCODE 486
0486:	FOR I := 2 TO L DO GET(LGO)	PASCODE 487
0487:	END;	PASCODE 488
0488:	TEXT,	PASCODE 489
0489:	REPL : BEGIN ADDWORD(W);	PASCODE 490
0490:	FOR I := 1 TO L DO	PASCODE 491
0491:	BEGIN ADDWORD(LGO^); GET(LGO) END	PASCODE 492
0492:	END;	PASCODE 493
0493:	LINK : BEGIN H := 0; W := LGO^;	PASCODE 494
0494:	WHILE L > 0 DO	PASCODE 495
0495:	BEGIN IF W.HALFWORD[0] = 0 THEN NEXTHALF;	PASCODE 496
0496:	IF L > 0	PASCODE 497
0497:	THEN BEGIN EXTNAME := W;	PASCODE 498
0498:	NEXTHALF; NEXTHALF;	PASCODE 499
0499:	SPACEFILL(EXTNAME.NAME);	PASCODE 500
0500:	WHILE W.TRAILER.T DO	PASCODE 501
0501:	BEGIN EXTC := EXTC + 1;	PASCODE 502
0502:	WITH EXT[EXTC], W.TRAILER DO	PASCODE 503
0503:	BEGIN NAME := EXTNAME.NAME;	PASCODE 504
0504:	ADDR := A;	PASCODE 505
0505:	POS := 2 - P;	PASCODE 506
0506:	NEXTHALF	PASCODE 507
0507:	END	PASCODE 508
0508:	END	PASCODE 509
0509:	END	PASCODE 510
0510:	END	PASCODE 511
0511:	END;	PASCODE 512
0512:	PRFX : BEGIN LGO^.ADDRESS.LOWER := 0; BLOCKNAME := LGO^.NAME;	PASCODE 513
0513:	SPACEFILL(BLOCKNAME);	PASCODE 514
0514:	GET(LGO); L := L - 1;	PASCODE 515
0515:	IF L >= 5	PASCODE 516
0516:	THEN BEGIN FOR I := 1 TO 5 DO	PASCODE 517
0517:	BEGIN CREATION[I] := LGO^.NAME;	PASCODE 518
0518:	SPACEFILL(CREATION[I]);	PASCODE 519
0519:	GET(LGO)	PASCODE 520
0520:	END;	PASCODE 521

0521:	PROCESSOR.NAME := CREATION[4];	PASCODE	522
0522:	FOREIGN := PROCESSOR.SHORTNAME <> 'PASCAL ';	PASCODE	523
0523:	L := L - 5	PASCODE	524
0524:	END;	PASCODE	525
0525:	FOR I := 1 TO L DO GET(LGO)	PASCODE	526
0526:	END	PASCODE	527
0527:	END	PASCODE	528
0528:	ELSE (* IGNORE THE TABLE *)	PASCODE	529
0529:	FOR I := 1 TO L DO GET(LGO)	PASCODE	530
0530:	END (* PROCESSTABLE *);	PASCODE	531
0531:		PASCODE	532
0532:	BEGIN (* PASS1 *)	PASCODE	533
0533:	PROCLENGTH := 0;	PASCODE	534
0534:	FOREIGN := TRUE;	PASCODE	535
0535:	CREATION[1] := ' ';	PASCODE	536
0536:	NEW(FSTTB);	PASCODE	537
0537:	LSTTB := FSTTB;	PASCODE	538
0538:	TBL := 0;	PASCODE	539
0539:	EXTC := 0;	PASCODE	540
0540:	WHILE NOT EOS(LGO) DO	PASCODE	541
0541:	PROCESSTABLE;	PASCODE	542
0542:	LSTTB^.NXTTB := NIL	PASCODE	543
0543:	END (* PASS1 *);	PASCODE	544
0544:		PASCODE	545
0545:		PASCODE	546
0546:		PASCODE	547
0547:		PASCODE	548
0548:	(* PASS2 - PRINT THE PSEUDO-COMPASS. *)	PASCODE	549
0549:		PASCODE	550
0550:		PASCODE	551
0551:	PROCEDURE PASS2;	PASCODE	552
0552:	TYPE DIGITPOSITION = SET OF 0..20;	PASCODE	553
0553:	VAR W,RELOCATION : WORD;	PASCODE	554
0554:	LC,LASTLC : INTEGER;	PASCODE	555
0555:	WORDTYPE : (WNAME,WDESCR,WCODE,WCONST,WVNAME,WVINFO);	PASCODE	556
0556:	ENDCODE,ENDCONST : INTEGER;	PASCODE	557
0557:		PASCODE	558
0558:		PASCODE	559
0559:	PROCEDURE NEXTWORD; (* GET NEXT REMEMBERED WORD *)	PASCODE	560
0560:	BEGIN W := TB^.T[TBI];	PASCODE	561
0561:	IF TB = LSTTB	PASCODE	562
0562:	THEN IF TBI = TBL	PASCODE	563
0563:	THEN TB := NIL	PASCODE	564
0564:	ELSE TBI := TBI + 1	PASCODE	565
0565:	ELSE IF TBI = TBLLEN	PASCODE	566
0566:	THEN BEGIN TB := TB^.NXTTB; TBI := 1 END	PASCODE	567
0567:	ELSE TBI := TBI + 1	PASCODE	568
0568:	END (* NEXTWORD *);	PASCODE	569
0569:		PASCODE	570
0570:		PASCODE	571
0571:	PROCEDURE PRINTTEXTTAB; (* PRINT ONE TEXT TABLE *)	PASCODE	572
0572:	VAR WC,WN : INTEGER;	PASCODE	573
0573:	FW : WORD;	PASCODE	574
0574:		PASCODE	575
0575:		PASCODE	576
0576:	PROCEDURE PRINTOCTAL(W : WORD; B : DIGITPOSITION);	PASCODE	577
0577:	(* PRINT A BLANK AFTER CERTAIN DIGIT POSITIONS *)	PASCODE	578
0578:	VAR I : INTEGER;	PASCODE	579
0579:	BEGIN (* PRINTOCTAL *)	PASCODE	580
0580:	IF 0 IN B THEN WRITE(' ');	PASCODE	581
0581:	FOR I := 1 TO 20 DO	PASCODE	582
0582:	BEGIN WRITE(CHR(W.DIGIT[I] + ORD('0')));	PASCODE	583
0583:	IF I IN B THEN WRITE(' ');	PASCODE	584
0584:	END	PASCODE	585
0585:	END (* PRINTOCTAL *);	PASCODE	586

0586:		PASCODE	587
0587:		PASCODE	588
0588:	PROCEDURE PRINTWORD; (* ADDRESS, OCTAL, ALFA, REAL, INTEGER *)	PASCODE	589
0589:	VAR SFNAME: ALFA;	PASCODE	590
0590:	BEGIN (* PRINTWORD *)	PASCODE	591
0591:	BEGINLINE;	PASCODE	592
0592:	WRITE(' ', LC: 6 OCT, ' ');	PASCODE	593
0593:	PRINTOCTAL(W,[4,8,12,16]);	PASCODE	594
0594:	WRITE(' ');	PASCODE	595
0595:	SFNAME := W.NAME;	PASCODE	596
0596:	SPACEFILL(SFNAME);	PASCODE	597
0597:	WRITE(SFNAME, ' ');	PASCODE	598
0598:	IF W.N <> 0	PASCODE	599
0599:	THEN BEGIN IF UNDEFINED(W.R)	PASCODE	600
0600:	THEN WRITE(' ': 22)	PASCODE	601
0601:	ELSE WRITE(W.R: 22);	PASCODE	602
0602:	IF ABS(W.N) <= MAXINT THEN WRITE(W.N: 17)	PASCODE	603
0603:	END;	PASCODE	604
0604:	WRITELN	PASCODE	605
0605:	END (* PRINTWORD *);	PASCODE	606
0606:		PASCODE	607
0607:		PASCODE	608
0608:	PROCEDURE PRINTCODE; (* INSTRUCTION DECOMPOSITION *)	PASCODE	609
0609:	VAR I,P,R,BL : INTEGER;	PASCODE	610
0610:	A : ALFA;	PASCODE	611
0611:	C : CHAR;	PASCODE	612
0612:	ADDNAME : ARRAY[0..3] OF ALFA;	PASCODE	613
0613:	SFNAME : ALFA;	PASCODE	614
0614:	EXECUTABLE,B0PK,OCTAL : BOOLEAN;	PASCODE	615
0615:	K : WORD;	PASCODE	616
0616:	BEGIN (* PRINTCODE *)	PASCODE	617
0617:	FOR I := 0 TO 3 DO ADDNAME[I] := 'NONE ';	PASCODE	618
0618:	FOR I := 1 TO EXTC DO WITH EXT[I] DO	PASCODE	619
0619:	IF ADDR = LC	PASCODE	620
0620:	THEN ADDNAME[POS] := NAME;	PASCODE	621
0621:	R := RELOCATION.REL[WN];	PASCODE	622
0622:	FOR I := 2 DOWNT0 0 DO	PASCODE	623
0623:	BEGIN R := R DIV 2;	PASCODE	624
0624:	IF ODD(R)	PASCODE	625
0625:	THEN ADDNAME[I] := 'PROC ';	PASCODE	626
0626:	END;	PASCODE	627
0627:	BL := 0; (* BLANK COUNTER *)	PASCODE	628
0628:	P := 0; (* PARCEL COUNTER *)	PASCODE	629
0629:	EXECUTABLE := TRUE;	PASCODE	630
0630:	OCTAL := FALSE;	PASCODE	631
0631:	BEGINLINE;	PASCODE	632
0632:	WRITE(' ', LC: 6 OCT, ' ');	PASCODE	633
0633:	IF FOREIGN	PASCODE	634
0634:	THEN BEGIN PRINTOCTAL(W,[4,8,12,16]);	PASCODE	635
0635:	SFNAME := W.NAME;	PASCODE	636
0636:	SPACEFILL(SFNAME);	PASCODE	637
0637:	WRITE(' ', SFNAME, ' ');	PASCODE	638
0638:	END;	PASCODE	639
0639:	REPEAT	PASCODE	640
0640:	IF BL <> 0 THEN WRITE(' ': BL);	PASCODE	641
0641:	BL := 0;	PASCODE	642
0642:	TOTALCOUNT := TOTALCOUNT + 1;	PASCODE	643
0643:	WITH W.QUARTER DO	PASCODE	644
0644:	BEGIN CODECOUNT[OP] := CODECOUNT[OP] + 1;	PASCODE	645
0645:	IF OP = 3 THEN TESTCOUNT[I] := TESTCOUNT[I] + 1;	PASCODE	646
0646:	IF ((OP IN [46B,63B..67B]) AND (I = 0)) THEN	PASCODE	647
0647:	NOOPS[EXECUTABLE] := NOOPS[EXECUTABLE] + 1	PASCODE	648
0648:	ELSE IF ((OP IN [60B..62B]) AND (I = 0)) THEN	PASCODE	649
0649:	BIGNOOPS[EXECUTABLE] := BIGNOOPS[EXECUTABLE] + 1;	PASCODE	650
0650:	IF OP <= 7 THEN EXECUTABLE := FALSE;	PASCODE	651

0651:	BOPK := (OP-50 IN [00B..12B,10B..12B,20B..22B]) AND	PASCODE	652
0652:	(I <> 0) AND (J = 0);	PASCODE	653
0653:	A := CODE[OP]	PASCODE	654
0654:	END;	PASCODE	655
0655:	IF OCTAL	PASCODE	656
0656:	THEN REPEAT WRITE(W.PARCEL[0]: 5 OCT);	PASCODE	657
0657:	P := P + 1;	PASCODE	658
0658:	W.N := W.N * PARCELSHIFT	PASCODE	659
0659:	UNTIL P > 3	PASCODE	660
0660:	ELSE FOR I := 1 TO 10 DO	PASCODE	661
0661:	BEGIN C := A[I];	PASCODE	662
0662:	IF C IN ['0'..'9','.']	PASCODE	663
0663:	THEN CASE C OF	PASCODE	664
0664:	'0' : WRITE(W.QUARTER.I: 1);	PASCODE	665
0665:	'1' : WRITE(W.QUARTER.J: 1);	PASCODE	666
0666:	'2' : WRITE(W.QUARTER.K: 1);	PASCODE	667
0667:	'3' : BEGIN IF (W.HALF.K = 0) AND (ADDNAME[P][1] <> '=')	PASCODE	668
0668:	THEN BEGIN WRITE('=X', ADDNAME[P]);	PASCODE	669
0669:	OCTAL := ADDNAME[P] = 'P.PEN ';	PASCODE	670
0670:	BL := 7	PASCODE	671
0671:	END	PASCODE	672
0672:	ELSE BEGIN WRITE(W.HALF.K: 6 OCT);	PASCODE	673
0673:	IF ADDNAME[P] = '=NONE '	PASCODE	674
0674:	THEN IF BOPK	PASCODE	675
0675:	THEN BEGIN K.N := W.HALF.K;	PASCODE	676
0676:	SPACEFILL(K.NAME);	PASCODE	677
0677:	WRITE(W.HALF.K: 7, ' ', K.NAME[8],	PASCODE	678
0678:	K.NAME[9], K.NAME[10], ' ');	PASCODE	679
0679:	BL := 0	PASCODE	680
0680:	END	PASCODE	681
0681:	ELSE BL := 13	PASCODE	682
0682:	ELSE IF ADDNAME[P] = '=PROC '	PASCODE	683
0683:	THEN BEGIN WRITE('(', PROCNAME:LPROCNAME, ')');	PASCODE	684
0684:	BL := BL + 10 - LPROCNAME	PASCODE	685
0685:	END	PASCODE	686
0686:	ELSE WRITE('=X', ADDNAME[P]);	PASCODE	687
0687:	END;	PASCODE	688
0688:	P := P + 1;	PASCODE	689
0689:	W.N := W.N * PARCELSHIFT	PASCODE	690
0690:	END ;	PASCODE	691
0691:	'4' : IF W.HALF.K >= 0	PASCODE	692
0692:	THEN WRITE('+')	PASCODE	693
0693:	ELSE BEGIN WRITE('-'); W.HALF.K := -W.HALF.K END;	PASCODE	694
0694:	'5' : WRITE(TEST[W.QUARTER.I]);	PASCODE	695
0695:	'6' : WRITE(' ');	PASCODE	696
0696:	'7' : WRITE(W.QUARTER.J * 10B + W.QUARTER.K: 2);	PASCODE	697
0697:	'8' : IF I = 10 THEN BL := BL + 3 ELSE WRITE(' ');	PASCODE	698
0698:	'9' : ;	PASCODE	699
0699:	'.' : BL := BL + 1;	PASCODE	700
0700:	END	PASCODE	701
0701:	ELSE WRITE(C)	PASCODE	702
0702:	END;	PASCODE	703
0703:	BL := BL + 10;	PASCODE	704
0704:	P := P + 1;	PASCODE	705
0705:	W.N := W.N * PARCELSHIFT	PASCODE	706
0706:	UNTIL P > 3;	PASCODE	707
0707:	WRITELN	PASCODE	708
0708:	END (* PRINTCODE *);	PASCODE	709
0709:		PASCODE	710
0710:		PASCODE	711
0711:	BEGIN (* PRINTTEXTTABLE *)	PASCODE	712
0712:	LC := W.ADDRESS.LOWER;	PASCODE	713
0713:	IF LC <> LASTLC	PASCODE	714
0714:	THEN BEGIN BEGINLINE; WRITELN END;	PASCODE	715
0715:	WC := W.HEADER.WCNT;	PASCODE	716

```

0716:      WN := 1;                                PASCODE 717
0717:      NEXTWORD;                                PASCODE 718
0718:      RELOCATION := W;                          PASCODE 719
0719:      WHILE WN < WC DO                         PASCODE 720
0720:      BEGIN NEXTWORD;                          PASCODE 721
0721:      CASE WORDTYPE OF                         PASCODE 722
0722:      WNAME : BEGIN FW := W;                   PASCODE 723
0723:      FW.ADDRESS.LOWER := 0;                   PASCODE 724
0724:      IF (FW.N = 4000000000000000000B) OR     PASCODE 725
0725:      (FW.N = 60000000000000000000B)         PASCODE 726
0726:      THEN (* MAIN PROGRAM *)                  PASCODE 727
0727:      BEGIN FW := W;                            PASCODE 728
0728:      NEXTWORD;                                PASCODE 729
0729:      PROCNAME := W.NAME;                      PASCODE 730
0730:      PROCTITLE;                               PASCODE 731
0731:      BEGINLINE;                               PASCODE 732
0732:      WRITE(' ', LC: 6 OCT, ' ');              PASCODE 733
0733:      PRINTOCTAL(FW,[1,4,10,14]);              PASCODE 734
0734:      WRITELN;                                  PASCODE 735
0735:      LC := LC + 1;                             PASCODE 736
0736:      WN := WN + 1;                             PASCODE 737
0737:      BEGINLINE;                               PASCODE 738
0738:      WRITE(' ', LC: 6 OCT, ' ', PROCNAME);    PASCODE 739
0739:      IF EPTNAME <> PROCNAME                    PASCODE 740
0740:      THEN WRITE(' ': 16, 'ENTRY = ', EPTNAME: 7); PASCODE 741
0741:      WRITELN;                                  PASCODE 742
0742:      WORDTYPE := WDESCR;                      PASCODE 743
0743:      END                                        PASCODE 744
0744:      ELSE WITH TB^.T[TBI].DESCRIPTOR DO       PASCODE 745
0745:      IF (PAD = 0) AND (COUNT = 0) AND (ENTRY > 1) AND PASCODE 746
0746:      (CONSTS >= ENTRY) AND (PMDINF >= CONSTS) PASCODE 747
0747:      THEN (* PROCEDURE OR FUNCTION *)        PASCODE 748
0748:      BEGIN PROCNAME := W.NAME;               PASCODE 749
0749:      PROCTITLE;                               PASCODE 750
0750:      BEGINLINE;                               PASCODE 751
0751:      WRITE(' ', LC: 6 OCT, ' ', PROCNAME);    PASCODE 752
0752:      IF EPTNAME <> PROCNAME                    PASCODE 753
0753:      THEN WRITE(' ': 16, 'ENTRY = ', EPTNAME: 7); PASCODE 754
0754:      WRITELN;                                  PASCODE 755
0755:      WORDTYPE := WDESCR                      PASCODE 756
0756:      END                                        PASCODE 757
0757:      ELSE (* PMD INFO ENTIRELY SUPPRESSED (P0) *) PASCODE 758
0758:      BEGIN FOREIGN := TRUE;                   PASCODE 759
0759:      PROCNAME := BLOCKNAME;                   PASCODE 760
0760:      PROCTITLE;                               PASCODE 761
0761:      PRINTCODE;                               PASCODE 762
0762:      ENDCODE := MAXINT;                       PASCODE 763
0763:      ENDCONST := MAXINT;                      PASCODE 764
0764:      WORDTYPE := WCODE                        PASCODE 765
0765:      END;                                     PASCODE 766
0766:      END;                                     PASCODE 767
0767:      WDESCR : WITH W.DESSCRIPTOR DO           PASCODE 768
0768:      BEGIN ENDCODE := CONSTS - 1;             PASCODE 769
0769:      ENDCONST := PMDINF - 1;                 PASCODE 770
0770:      BEGINLINE;                               PASCODE 771
0771:      WRITE(' ', LC: 6 OCT, ' ');              PASCODE 772
0772:      PRINTOCTAL(W,[1,5,10,15]);              PASCODE 773
0773:      IF NOT P THEN WRITE(' (PMD OFF)');       PASCODE 774
0774:      IF S THEN WRITE(' (PMD SUPPRESSED)');    PASCODE 775
0775:      IF F THEN WRITE(' (MODULE IS A FUNCTION)'); PASCODE 776
0776:      WRITELN;                                  PASCODE 777
0777:      IF LC >= ENDCODE                         PASCODE 778
0778:      THEN WORDTYPE := WCONST                  PASCODE 779
0779:      ELSE WORDTYPE := WCODE;                  PASCODE 780
0780:      BEGINLINE; WRITELN                      PASCODE 781

```

0781:	END;	PASCODE	782
0782:	WCODE : BEGIN PRINTCODE;	PASCODE	783
0783:	IF LC >= ENDCODE THEN WORDTYPE := WCONST;	PASCODE	784
0784:	IF LC >= ENDCONST THEN WORDTYPE := WVNAME	PASCODE	785
0785:	END;	PASCODE	786
0786:	WCONST : BEGIN IF LC = ENDCODE + 1	PASCODE	787
0787:	THEN BEGIN BEGINLINE; WRITELN END;	PASCODE	788
0788:	PRINTWORD;	PASCODE	789
0789:	IF LC >= ENDCONST THEN WORDTYPE := WVNAME	PASCODE	790
0790:	END;	PASCODE	791
0791:	WVNAME : BEGIN IF W.N <> 0	PASCODE	792
0792:	THEN BEGIN IF LC = ENDCONST + 1	PASCODE	793
0793:	THEN BEGIN BEGINLINE; WRITELN END;	PASCODE	794
0794:	BEGINLINE;	PASCODE	795
0795:	WRITE(' ', LC: 6 OCT, ' ', W.NAME)	PASCODE	796
0796:	END;	PASCODE	797
0797:	WORDTYPE := WVINFO	PASCODE	798
0798:	END;	PASCODE	799
0799:	WVINFO : BEGIN WRITELN(' : ', TYP[W.INFO.VTYPE DIV 2], ' ',	PASCODE	800
0800:	W.INFO.VADDR: 6 OCT);	PASCODE	801
0801:	WORDTYPE := WVNAME	PASCODE	802
0802:	END	PASCODE	803
0803:	END;	PASCODE	804
0804:	LC := LC + 1;	PASCODE	805
0805:	WN := WN + 1	PASCODE	806
0806:	END;	PASCODE	807
0807:	IF LC > PROCLENGTH THEN PROCLENGTH := LC;	PASCODE	808
0808:	LASTLC := LC	PASCODE	809
0809:	END (* PRINTTEXTTABLE *);	PASCODE	810
0810:		PASCODE	811
0811:		PASCODE	812
0812:	PROCEDURE PRINTREPLTABLE; (* PRINT REPLICATION INFO *)	PASCODE	813
0813:	VAR WC : INTEGER;	PASCODE	814
0814:	FW : WORD;	PASCODE	815
0815:	BEGIN (* PRINTREPLTABLE *)	PASCODE	816
0816:	WC := W.HEADER.WCNT;	PASCODE	817
0817:	NEXTWORD;	PASCODE	818
0818:	FW := W;	PASCODE	819
0819:	NEXTWORD;	PASCODE	820
0820:	WHILE WC > 0 DO	PASCODE	821
0821:	BEGIN BEGINLINE; WRITELN;	PASCODE	822
0822:	BEGINLINE;	PASCODE	823
0823:	WITH FW.REPL1, W.REPL2 DO	PASCODE	824
0824:	BEGIN IF BSIZE = 0 THEN BSIZE := 1;	PASCODE	825
0825:	WRITE(' ': 9, 'COPY ', BSIZE: 1, ' WORD');	PASCODE	826
0826:	IF BSIZE <> 1 THEN WRITE('S');	PASCODE	827
0827:	IF COUNT > 1	PASCODE	828
0828:	THEN WRITE(' ', COUNT: 1, ' TIMES');	PASCODE	829
0829:	WRITELN(' FROM ADDRESS ', SADDR: 6 OCT,	PASCODE	830
0830:	' TO ADDRESS ', DADDR: 6 OCT)	PASCODE	831
0831:	END;	PASCODE	832
0832:	IF WC > 2	PASCODE	833
0833:	THEN BEGIN NEXTWORD;	PASCODE	834
0834:	IF WC > 1 THEN NEXTWORD	PASCODE	835
0835:	END;	PASCODE	836
0836:	WC := WC - 2	PASCODE	837
0837:	END;	PASCODE	838
0838:	LASTLC := -1	PASCODE	839
0839:	END (* PRINTREPLTABLE *);	PASCODE	840
0840:		PASCODE	841
0841:		PASCODE	842
0842:	BEGIN (* PASS2 *)	PASCODE	843
0843:	IF FOREIGN	PASCODE	844
0844:	THEN BEGIN PROCNAME := BLOCKNAME;	PASCODE	845
0845:	PROCTITLE;	PASCODE	846

0846:	ENDCODE := MAXINT;	PASCODE	847
0847:	ENDCONST := MAXINT;	PASCODE	848
0848:	WORDTYPE := WCODE	PASCODE	849
0849:	END	PASCODE	850
0850:	ELSE WORDTYPE := WNAME;	PASCODE	851
0851:	IF TBL <> 0	PASCODE	852
0852:	THEN BEGIN LC := 0;	PASCODE	853
0853:	LASTLC := 0;	PASCODE	854
0854:	TB := FSTTB;	PASCODE	855
0855:	TBI := 1;	PASCODE	856
0856:	WHILE TB <> NIL DO	PASCODE	857
0857:	BEGIN NEXTWORD;	PASCODE	858
0858:	IF W.HEADER.TAG DIV 100B = TEXT	PASCODE	859
0859:	THEN PRINTTEXTTABLE	PASCODE	860
0860:	ELSE PRINTREPLTABLE	PASCODE	861
0861:	END	PASCODE	862
0862:	END;	PASCODE	863
0863:	WHILE FSTTB <> NIL DO	PASCODE	864
0864:	BEGIN TB := FSTTB^.NXTTB;	PASCODE	865
0865:	DISPOSE(FSTTB);	PASCODE	866
0866:	FSTTB := TB	PASCODE	867
0867:	END;	PASCODE	868
0868:	INDEXTHIS^.LENGTH := PROCLENGTH	PASCODE	869
0869:	END (* PASS2 *);	PASCODE	870
0870:		PASCODE	871
0871:		PASCODE	872
0872:		PASCODE	873
0873:		PASCODE	874
0874:	(* WRITECOUNTS - WRITE INSTRUCTION COUNTS *)	PASCODE	875
0875:		PASCODE	876
0876:		PASCODE	877
0877:	PROCEDURE WRITECOUNTS;	PASCODE	878
0878:	VAR I,J: INTEGER;	PASCODE	879
0879:		PASCODE	880
0880:		PASCODE	881
0881:	PROCEDURE WRITEOP(O: INTEGER); (* WRITE OPCODE MNEMONIC *)	PASCODE	882
0882:	VAR I: INTEGER;	PASCODE	883
0883:	A: ALFA;	PASCODE	884
0884:	C: CHAR;	PASCODE	885
0885:	BEGIN (* WRITEOP *)	PASCODE	886
0886:	WRITE(O: 2 OCT, ' ');	PASCODE	887
0887:	A := CODE[O];	PASCODE	888
0888:	FOR I := 1 TO 10 DO	PASCODE	889
0889:	BEGIN C := A[I];	PASCODE	890
0890:	IF C IN ['0'..'9', '.']	PASCODE	891
0891:	THEN CASE C OF	PASCODE	892
0892:	'0' : WRITE('I');	PASCODE	893
0893:	'1' : WRITE('J');	PASCODE	894
0894:	'2' : WRITE('K');	PASCODE	895
0895:	'3' : WRITE('K');	PASCODE	896
0896:	'4' : WRITE('+');	PASCODE	897
0897:	'5' : WRITE('--');	PASCODE	898
0898:	'6' : WRITE(' ');	PASCODE	899
0899:	'7' : WRITE('JK');	PASCODE	900
0900:	'8' : WRITE(' ');	PASCODE	901
0901:	'9' : ;	PASCODE	902
0902:	'.' : WRITE(' ');	PASCODE	903
0903:	END	PASCODE	904
0904:	ELSE WRITE(C)	PASCODE	905
0905:	END	PASCODE	906
0906:	END (* WRITEOP *);	PASCODE	907
0907:		PASCODE	908
0908:		PASCODE	909
0909:		PASCODE	910
0910:	PROCEDURE WRITETEST(T : INTEGER);	PASCODE	911

```

0911:                (* WRITE AN X-REGISTER TEST MNEMONIC *)           PASCODE  912
0912:    BEGIN (* WRITETEST *)                                         PASCODE  913
0913:    WRITE('03', T: 1, ' ', TEST[T], ' XI,K')                       PASCODE  914
0914:    END (* WRITETEST *);                                           PASCODE  915
0915:                                                                    PASCODE  916
0916:                                                                    PASCODE  917
0917:    PROCEDURE WRITEC(C : INTEGER); (* WRITE AN OPCODE COUNT *)     PASCODE  918
0918:    BEGIN (* WRITEC *)                                              PASCODE  919
0919:    WRITE(C: 7, C*100/TOTALCOUNT: 9:2)                             PASCODE  920
0920:    END (* WRITEC *);                                              PASCODE  921
0921:                                                                    PASCODE  922
0922:                                                                    PASCODE  923
0923:    BEGIN (* WRITECOUNTS *)                                       PASCODE  924
0924:    EJECT;                                                           PASCODE  925
0925:    TITLEMESSAGE := 'DECODE STATISTICS                               '; PASCODE  926
0926:    LTITLEMESSAGE := 17;                                           PASCODE  927
0927:    LPROCNAME := 0;                                               PASCODE  928
0928:    CREATION[1] := '                                             '; PASCODE  929
0929:    BEGINLINE; WRITELN;                                           PASCODE  930
0930:    BEGINLINE;                                                    PASCODE  931
0931:    WRITELN('          OPCODE          COUNT  PERCENT',           PASCODE  932
0932:    '          OPCODE          COUNT  PERCENT');                   PASCODE  933
0933:    BEGINLINE; WRITELN;                                           PASCODE  934
0934:    FOR I := 0 TO 37B DO                                           PASCODE  935
0935:    BEGIN BEGINLINE; J := I + 40B;                                  PASCODE  936
0936:    WRITE(' ': 9); WRITEOP(I); WRITEC(CODECOUNT[I]);             PASCODE  937
0937:    WRITE(' ': 8); WRITEOP(J); WRITEC(CODECOUNT[J]);             PASCODE  938
0938:    WRITELN                                                         PASCODE  939
0939:    END;                                                            PASCODE  940
0940:    BEGINLINE; WRITELN;                                           PASCODE  941
0941:    FOR I := 0 TO 3 DO                                             PASCODE  942
0942:    BEGIN BEGINLINE; J := I + 4;                                    PASCODE  943
0943:    WRITE(' ': 9); WRITETEST(I); WRITEC(TESTCOUNT[I]);          PASCODE  944
0944:    WRITE(' ': 8); WRITETEST(J); WRITEC(TESTCOUNT[J]);          PASCODE  945
0945:    WRITELN;                                                       PASCODE  946
0946:    END;                                                            PASCODE  947
0947:    BEGINLINE; WRITELN;                                           PASCODE  948
0948:    BEGINLINE; WRITELN;                                           PASCODE  949
0949:    BEGINLINE;                                                    PASCODE  950
0950:    WRITE(' ': 9, 'INACCESSABLE 1 PARCEL NO-OPS =');              PASCODE  951
0951:    WRITEC(NOOPS[FALSE]); WRITELN(' PERCENT');                    PASCODE  952
0952:    BEGINLINE;                                                    PASCODE  953
0953:    WRITE(' ': 9, 'INACCESSABLE 2 PARCEL NO-OPS =');              PASCODE  954
0954:    WRITEC(BIGNOOPS[FALSE]); WRITELN(' PERCENT');                 PASCODE  955
0955:    BEGINLINE; WRITELN;                                           PASCODE  956
0956:    BEGINLINE;                                                    PASCODE  957
0957:    WRITE(' ': 9, 'ACCESSABLE 1 PARCEL NO-OPS =');                PASCODE  958
0958:    WRITEC(NOOPS[TRUE]); WRITELN(' PERCENT');                     PASCODE  959
0959:    BEGINLINE;                                                    PASCODE  960
0960:    WRITE(' ': 9, 'ACCESSABLE 2 PARCEL NO-OPS =');                PASCODE  961
0961:    WRITEC(BIGNOOPS[TRUE]); WRITELN(' PERCENT');                  PASCODE  962
0962:    BEGINLINE; WRITELN;                                           PASCODE  963
0963:    BEGINLINE; WRITELN;                                           PASCODE  964
0964:    BEGINLINE;                                                    PASCODE  965
0965:    WRITE(' ': 9, 'TOTAL INSTRUCTIONS                               ='); PASCODE  966
0966:    WRITEC(TOTALCOUNT); WRITELN(' PERCENT');                     PASCODE  967
0967:    END (* WRITECOUNTS *);                                       PASCODE  968
0968:                                                                    PASCODE  969
0969:                                                                    PASCODE  970
0970:                                                                    PASCODE  971
0971:                                                                    PASCODE  972
0972:    (* WRITEINDEX - WRITE INDEX TO THE LISTING *)                  PASCODE  973
0973:                                                                    PASCODE  974
0974:                                                                    PASCODE  975
0975:    PROCEDURE WRITEINDEX;                                          PASCODE  976

```

```

0976:  VAR SCAN : PINDEXREC;                                PASCODE  977
0977:      TOTAL : INTEGER;                                PASCODE  978
0978:  BEGIN (* WRITEINDEX *)                               PASCODE  979
0979:  SCAN := INDEXBASE^.NEXT;                              PASCODE  980
0980:  TOTAL := 0;                                           PASCODE  981
0981:  EJECT;                                                PASCODE  982
0982:  TITLEMESSAGE := 'DECODE INDEX                        '; PASCODE  983
0983:  LTITLEMESSAGE := 12;                                   PASCODE  984
0984:  LPROCNAME := 0;                                       PASCODE  985
0985:  CREATION[1] := '          ';                          PASCODE  986
0986:  BEGINLINE; WRITELN;                                    PASCODE  987
0987:  BEGINLINE;                                             PASCODE  988
0988:  WRITELN('          MODULE          PAGE    LENGTH'); PASCODE  989
0989:  BEGINLINE; WRITELN;                                    PASCODE  990
0990:  WHILE SCAN <> NIL DO                                    PASCODE  991
0991:    WITH SCAN^ DO                                        PASCODE  992
0992:      BEGIN BEGINLINE;                                  PASCODE  993
0993:        WRITE('          ', NAME, PAGE: 6, '          ', LENGTH: 6 OCT); PASCODE  994
0994:        TOTAL := TOTAL + LENGTH;                        PASCODE  995
0995:        IF ENTRYNAME <> NAME                            PASCODE  996
0996:          THEN WRITE('          (', ENTRYNAME: 7, ')'); PASCODE  997
0997:        WRITELN;                                         PASCODE  998
0998:        SCAN := NEXT                                     PASCODE  999
0999:      END;                                               PASCODE 1000
1000:  BEGINLINE;                                             PASCODE 1001
1001:  WRITELN(' ': 29, '=====');                          PASCODE 1002
1002:  BEGINLINE;                                             PASCODE 1003
1003:  WRITELN(' ': 9, 'TOTAL', ' ': 15, TOTAL: 6 OCT)      PASCODE 1004
1004:  END (* WRITEINDEX *);                                  PASCODE 1005
1005:                                                         PASCODE 1006
1006:                                                         PASCODE 1007
1007:                                                         PASCODE 1008
1008:                                                         PASCODE 1009
1009:  BEGIN (* PASCODE *)                                    PASCODE 1010
1010:  RESET(LGO);                                           PASCODE 1011
1011:  LINELIMIT(OUTPUT,MAXINT);                              PASCODE 1012
1012:  DATE(TODAY);                                          PASCODE 1013
1013:  TABLES := [PIDL-30B, ENTR-30B, TEXT-30B, REPL-30B, LINK-30B, PRFX-30B]; PASCODE 1014
1014:  TIME(NOW);                                            PASCODE 1015
1015:  PAGECOUNT := 0;                                      PASCODE 1016
1016:  EJECT;                                                PASCODE 1017
1017:  NEW(INDEXBASE);                                        PASCODE 1018
1018:  INDEXBASE^.ZNAME := '          ';                     PASCODE 1019
1019:  ZEROFILL(INDEXBASE^.ZNAME);                          PASCODE 1020
1020:  INDEXBASE^.NEXT := NIL;                               PASCODE 1021
1021:  INDEXCOUNT := 0;                                    PASCODE 1022
1022:  WHILE NOT EOF(LGO) DO                                  PASCODE 1023
1023:    BEGIN PASS1;                                        PASCODE 1024
1024:    PASS2;                                             PASCODE 1025
1025:    GETSEG(LGO)                                        PASCODE 1026
1026:    END;                                               PASCODE 1027
1027:  IF PAGECOUNT > 10 THEN WRITECOUNTS;                PASCODE 1028
1028:  IF INDEXCOUNT > 5 THEN WRITEINDEX                   PASCODE 1029
1029:  END (* PASCODE *).                                    PASCODE 1030

```

THAT'S ALL FOLKS! LINES: 1029 CHARACTERS: 88494