

**ELLIOTT**

**903**

**OPERATING**

**NOTES**

**16/10/2014**



ACD UTILITIES

\*\*BINCOP.

- 8. Read & Store. Even Parity
- 12. Read & Store. Track 8 Blank
- 11. Read & Store. Otherwise.
- 9. Read & Store. Continue after 2 inches Blank.
- 10. Read & Check then Punch.

0000.001 Parity Error  
1000.000 Track 8 Not Blank.  
1111.111 Store Full.  
0000.010 Reader & Store Disagree.

\*\*TELECOP.

- 8. Read & Store. First Tape.
- 9. Read & Store. Subsequent Tape.
- 10. Read & Check then Punch Clean. First Tape.
- 11. Read and Check then Punch Unclean. First Tape.
- 9. Read and Check then Punch. Subsequent Tape.

\*\*DO-ALL 4.

- 8. Convert to Legible Tape.
- 9. Select 920 Telecode
- 10. Select 900 Telecode.
- 19. Select 903 Telecode for Algol.
- 11. Select Clean.
- 12. Select Unclean.
- 16. Read and Punch Telecode. First Tape.
- 17. Read and Punch Telecode. Subsequent Tape.
- 18. Punch <H>
- 20. Read and Form Sumchecks. Even Parity.
- 21. Read & Form Sumchecks. Otherwise.
- 17. Read and Form Sumchecks. Continue after 2 inches Blank.
- 22. Read & Punch (QCOPY). Check Sumchecks.
- 23. Read & Check Sumchecks.

CH I/O ERROR 1. Parity Error.  
CH I/O ERROR 2. First Character not <N>.  
CH I/O ERROR 3. Illegal Character.  
CH I/O ERROR 4. Over 120 Characters on Line.  
QCOPY ERROR 1. Parity Error.  
QCOPY ERROR 2. Sumchecks Disagree.

\*\*BOWDLER

CAM CAD

CSE CSI CII CIE CLE CLI DSE DSI DIE DII DLE DLI String  
CHI CHE DHI DHE PNL PHT PSP DNL DHT DSP

INS  
String^%,

RTS String                    RTI String                    RTL String                    RTH  
String ^%,                    String^%,                    String^%,                    String^%,

REP Digit                    END                    CAN Digit  
WAI                    WTS                    COP

8. Read Steering Tape. Errors in 920 Telecode.
13. Read Steering Tape. Errors in 900 Telecode.
9. Read and Edit Main Tape.
10. Read & Edit Main Tape. Continue after Wait.
11. Output Instruction Count.

A; <H> in REP.  
 B: Not All REPs Cancelled.  
 C: <H> in Comment.  
 D: <H> in Command.  
 E: Unrecognized Command.  
 F: More than 120 Characters in Line.  
 G: Telecode Commands not at Start.  
 H: Nested REP.  
 I: CAN inside REP.  
 J: COP inside REP.  
 K: END outside REP.  
 L: Empty REP.  
 M: First Instruction in REP not Copy.  
 N: REP Does Not Advance.  
 O: REP or CAN Label Error.  
 P: CAN Label Not Declared or Cancelled.  
 Q: REP Label Active in Other REP.  
 R: More than 32 REPs Active.  
 S: Spurious Character in Command.  
 T: String Expected.  
 U: More Than 3 Characters in Escape Sequence.  
 V: Spurious Character After ^%.  
 W: <H> Encountered While Matching.  
 X: Command Buffer Full.

\*\*TATDOC

15. Select 900 Telecode.
16. Select 920 Telecode.

8. Read & Doc Program or General Tape. First Tape.
9. Read & Doc Program or General Tape. Subsequent Tape.
10. Punch end of Final Page & <H>.
11. Read & Doc Label List. SIR or SAP pass 1.
9. Read & Doc Label List. SAP pass 2.

12. Read in Top & Tail. <N>\*\*\*\*\*<N>.
13. As 8 but Punch Top & Tail.
14. As 11 but Punch Top & Tail.

CH I/O ERROR 1. Parity Error.  
 CH I/O ERROR 2. First Character not <N>.  
 CH I/O ERROR 3. Illegal Character.  
 DOCUM. ERROR 1. Illegal Entry Point.  
 DOCUM. ERROR 2. Top & Tail Wrong.  
 DOCUM. ERROR 3. Title Wrong.  
 DOCUM. ERROR 4. Over 74 Characters on Line.  
 DOCUM. ERROR 5. <H> Wrong.

\*\*DUMP, CLEAR STORE

Self Triggering.

\*\*VERIFY

&17501. Verify.

&Address &Word on Tape &Word in Store.

EL. Wrong Loader.  
 EF. Format Error.  
 ES. Sumcheck Error.  
 EA. Users Program Corrupted by Verify.

\*\*AMEND

Title. Comment. Absolute patch. Skip.  
Absolute & Relative Instruction. Integer. Octal.

0000.001 Parity Error.  
0000.111 Any Other Error.

\*\*STORE PRINT

Unsigned Address. Optional Unsigned Address. Letter.

O. Instruction. I. Integer.  
F. Fraction. B. Octal.

\*\*GENSORT

8. Read & Store Alphabet Tape. <N>\*\*\*\*\*<N><H>
9. Read & Store Unsorted Tape. First Tape.
10. Read & Store Unsorted Tape. Subsequent Tape.
11. Read Sorted Tape. Merge with Store & Punch.
12. Punch Remainder of Store.

0000.001 First Character Not <N>.  
0000.010 Character Not in Alphabet.  
0000.011 Character in Alphabet Twice.  
0000.100 Over 63 Characters in Alphabet.  
0000.101 <H> or <N> Wrong in Alphabet.  
0000.110 Character Store Full.

\*\*ALGCON

8. Read & Punch First Tape, for 903 ALGOL.
9. Read & Punch First Tape, for 920 ALGOL.
10. Read & Punch First Tape, for 503 ALGOL.
11. Read & Punch Subsequent Tape.
12. Punch <H>

AJH UTILITIES

\*\*VT220.

8. Input text from VT2220.
9. Read text from paper tape and display page by page on VT220.
10. Produce 6" of runout.

Input

ERASE deletes previous character if any on current line.  
Control-X discards current line.  
RETURN or LINEFEED (control-J) each insert a newline sequence.  
Halt code (control-T) terminates input and outputs buffered input to paper tape.

Output

Type RETURN to advance to next screen.



ALGOL (903 Algol, MASD Algol, Hunter Algol, AJH Algol)

TRANSLATOR (TAPE 1)

- 8 Normal translate.
- 9 Continue translation after halt (in any mode).
- 10 Report mode. (No intermediate code is produced).
- 11 Normal translate, include check functions.
- 12 Translate in library mode.
- 13 Translate in library mode plus check functions.
- 14 Translate in normal mode and give reports.
- 16 Freeze name list and punch out a copy of the translator.
- 17 Enable call-by-name extensions (AJH ALGOL only).
- 18 Disable call-by-name extensions (AJH ALGOL only).

INTERPRETER (TAPE 2)

- 8 Normal, load translated program.
- 9 Continue after halt.
- 10 Run program.
- 11 Load additional code procedures relocatable binary tape.
- 12 Reload the standard library (TAPE 3) after entry at 13.
- 13 Load a large program overwriting library.
- 14 Dump the interpreter and any loaded code.

Load and Go Systems.

- 8 Normal translation.
- 9 Continue translation after halt.
- 10 Run program.
- 11 Load and go translation with check functions enabled and reports.
- 12 Load and go translation in library mode.
- 13 Load and go translation in library mode, with check functions and reports.
- 14 Load additional code procedures relocatable binary tape.
- 15 Re-establish built-in library (903 Algol only).
- 16 Translate in report mode. (No executable code is produced).
- 17 Translate to relocatable binary code format paper tape.
- 18 Re-establish the library (HUNTER & AJH Algol only).
- 19 Checksum the store and lock the system (HUNTER & AJH Algol only)
- 23 Enable call-by-name extensions (AJH Algol only).
- 24 Disable call-by-name extensions (AJH Algol only).
- 25 Freeze current translator name list (i.e., will not be overwritten by next entry at 8 or 11).

Large Program Systems.

- 8 Load an ALGOL program maintaining existing library and code procedures.
- 9 Load an ALGOL program maintaining existing library only.
- 10 Load an ALGOL program overwriting existing library and code procedures.
- 11 Load a code procedure maintaining existing code procedures.
- 12 Load a code procedure overwriting existing code procedures.
- 13 Load the library overwriting existing library and code procedures.
- 14 Dump a copy of the interpreter, library and code procedures.

Compile time errors

1. read misplaced.
2. print misplaced.
3. Constant or expression in read list.
4. Wrong delimiter in switch declaration.
5. Illegal actual parameter.
6. Too many parameters to a procedure.

7. Illegal number.
8. Integer constant too big.
9. Two statements in the same block are prefixed by the same label.
10. Identifier or constant not as expected.
11. Letter, digit or subscript ten misused.
12. true or false follows an identifier or constant.
13. comment does not follow semicolon or begin.
14. An internal error.
15. Unrecognised basic symbol, e.g., caused by fred.
16. No assignment within a type procedure to procedure name.
17. Identifier in value or spec. part is not a formal parameter.
18. Use of undeclared identifier.
19. Illegal symbol.
20. Non procedure identifier used as a statement.
21. "!=" omitted from for clause.
22. Illegal use of label name.
23. Inadmissible array declaration.
24. <switch name> not an actual parameter nor preceded by goto.
25. Non type procedure as function designator.
26. switch misplaced.
27. Declaration without identifier.
28. "!=" preceded by a constant or used inside an expression.
29. ":" in type or switch declaration misused.
30. Adjacent delimiters inadmissible.
31. Constant before "!=" or "[", or constant or string name in read list.
32. Item other than a non-type procedure used as a statement.
33. Identifier or constant follows a closing round or square bracket.
34. Relation on each side of a simple arithmetic expression.
35. Illegal statement, delimiter misused.
36. Declaration starts incorrectly.
37. Error between for and "!=".
38. Missing array or switch name, or "[" misplaced.
39. ";" misused in array declaration.
40. end misused.
41. Local identifier used in array bounds.
42. goto follows an identifier or constant.
43. Wrong for clause preceding do.
44. for misused.
45. Misused boolean constant.
46. Assignment to procedure identifier outside procedure body.
47. real, integer or boolean misplaced.
48. Identifier declared twice in same block head.
49. Blank parameter.
50. No begin at start of program.
51. Wrong number of subscripts or parameters.
52. "!=" appears in actual parameter list.
53. Statement ends incorrectly.
54. Declaration follows statement.
55. ":", goto or for used in expression.
56. Illegal parameter comment or ")" precedes identifier.
57. Wrong use of delimiter.
58. Relational or logical operator used as an arithmetic operator.
59. Illegal use of logical operator.
60. Omission or error precedes begin, or begin follows "!=".
61. "(" misplaced or missing procedure name.
62. Function designator as designational expression.
63. Misplaced declarator.
64. Subscripted variable as statement.
65. Illegal specifier.
66. Misused comma or colon in an expression.
67. if misused or used in read or print list.
68. if used in type declaration.

69. Corresponding if has been omitted, or conditional expression without an else.
70. Corresponding then missing.
71. Illegal character in inner string, or missing close quotes in previous string.
72. array misplaced.
73. "[" not preceded by an identifier.
74. Unmatched "]"
75. Upper bound missing in an array declaration.
76. Illegal type declaration.
77. Illegal array list.
78. Corresponding for missing.
79. A jump is made to a label declared, but not placed in the block that ends here. Or, for Hunter Algol, identifier used as procedure actual argument in procedure call within current block not declared.
80. step, until or while misused in a for statement.
81. Misused ")" other than in an expression.
82. ")" misplaced or unmatched
83. Program too complex, i.e., some statement is too complicated.
84. Wrong delimiter after procedure statement.
85. Program too large, i.e., it contains too many names, labels, constants or switches.
86. Error before procedure.
87. Repeated formal parameter.
88. Wrong formal parameter delimiter.
89. In a call of a formal procedure, one of its parameters is ln or exp.
90. Wrong delimiter in value or specification part.
91. Input buffer overflow, i.e., more than 120 characters in a line.
92. Formal parameter has not appeared in the specification part.
93. Declaration terminated by end or containing begin.
94. A formal parameter which is a switch, string or procedure is called by value.
95. Switch designator has more than one subscript.
96. Wrong for clause.
97. then misused.
98. Illegal character or parity error. The character is replaced by the back-arrow character in the displayed line but "↑" is not printed beneath it
99. Current use of identifier inconsistent with previous uses.
100. Conditional expression needs parentheses.
101. Wrong delimiter after procedure identifier in procedure declaration.
102. No ";" between formal parameter part and value or specification part.
103. Commas or colons wrong in array bounds.
104. div used with a real argument.
105. Illegal parameter delimiter after a string.
106. Integer labels not allowed.
107. Recursive function calls not allowed.
108. An actual parameter which is a procedure has one of its parameters called by value. (Restriction removed in Hunter ALGOL).
109. Constant should not be used in procedure heading.
110. Wrong specification part.
111. Different number of parameters from previous use of formal procedure or wrong number of subscripts.
112. Mixed type in multiple assignment.

#### Load time errors

FA Misread or mispunched paper tape.  
FC Two procedures (at least one of which is a library procedure) have the same name.  
FD As FA.  
FE Program too large to load.  
FF As FA.  
FU Missing library procedure.

#### Run time errors

1. Parameter mismatch
2. Space overflow, too much claim on store.
3. Integer overflow.
4. Jump error, i.e., switch subscript outside range.
5. Subscript error.
6. Illegal symbol inside inner string quotes.
7. Attempt to output non-standard floating point number.
8. Illegal character or ' found when reading a number.
9. Real overflow.
10. Invalid argument to sin or cos.
11. Negative argument to sqrt.
12. Argument > 40 for exp.
13. Negative argument for ln.
14. Illegal character on data tape.
15. Parity error on data tape.
16. Input buffer overflow, i.e., longer than 120 characters.
17. Numeric character encountered before ' on instring.
18. Illegal form of number, eg two decimal points.
19.  $A^B$  with A and B real and  $A < 0$ .
20. Corrupt program, possibly because of assembly coded bit.
21. Attempt to assign to formal parameter which is a constant.
22. Range of array subscript bounds is negative.
23. instring or outstring error.
24. Attempt to jump to label in inner block.
25. Translator version incompatible with run-time system.

## 903 FORTRAN

### Tape 1

- 8 Translate to SIR code.
- 9 Continue translation after halt.
- 10 Report Mode (check for errors only).

### Tape 2

- 8 Load SIR code (from Translation phase) for batch mode.
- 9 Continue after wait.
- 10 Indicate program complete.
- 11 Run the program.
- 12 Load for batch mode, display store map.
- 13 Load SIR code (from Translation phase) for relocatable mode.
- 14 Load relocatable program tape.
- 15 Load an extra relocatable binary (user's library) tape.

### LG

- 8 Read program.
- 9 Continue translation after halt.
- 10 Indicate program complete.
- 11 Run the program.
- 12 Read program and display store map.
- 15 Load an extra relocatable binary (user's library) tape.
- 16 Report mode (scan for errors only).

### LP

- 9 Continue after halt.
- 10 Indicate program complete.
- 11 Run the program.
- 13 Read program.
- 14 Load relocatable program tape.
- 15 Load separately compiled code procedures.
- 16 Report mode (scan for errors only).

### Compile time errors

1. Unacceptable form to the left of = sign in an arithmetic statement.
- 2.
3. Two successive operators in an arithmetic expression.
- 4.
5. Different number of left and right parentheses.
6. Subscripted variable not declared in DIMENSION statement.
7. Illegal form of subscript.
8. An unsubscripted array name in an arithmetic statement.

### In a statement DO n I = m1, m2, m3 (errors 9 to 17)

9. n is omitted or not an acceptable statement number.
10. I is omitted or not an unsubscripted integer variable
11. There is an impermissible number of m's
12. One of the m's is in impermissible form
- 13.
14. DO statements have intersecting loops.
15. A DO terminates with a GO TO or IF statement.
16. N has not been found when END is reached.
17. Format error in a DO statement.
  
18. A number has been found where a variable is expected.
19. No variable found where one expected.
20. A statement number has more than 5 digits.
21. An integer or real constant is out of range.
22. A CALL statement has an unacceptable format.
23. A FUNCTION or SUBROUTINE statement is unacceptable.
24. The word FOTRAN has not been preceded by a CODE statement,
25. An error in a GO TO or IF statement.

26. A misspelt or otherwise unrecognizable statement.
27. Statement too long or too complex to compile.
28. Program too long or too complex to compile.
29. Error in FORMAT statement.
- 30.
31. Error in DIMENSION statement.
32. Error in COMMON statement.
33. A variable has been declared twice in a COMMON list.
- 34.
- 35.
- 36.
- 37.
38. In a DIMENSION statement, an array exceeds 8192 elements.
39. A variable has been declared twice in a DIMENSION statement.
40. A subprogram has more than 18 parameters.
41. A continuation line has been used other than after a GLOBAL, COMMENT or FORMAT statement.
42. Too many subprograms declared in a GLOBAL statement.
43. Error in a READ or WRITE statement.
44. A FUNCTION or SAUBROUTINE statement has appeared in a sub-program.
45. Error in a GLOBAL statement.
46. Too many variables have been used in a program or subprogram.
47. A subprogram does not contain a RETURN statement, or, RETURN has occurred outside a subprogram.
- 48.
49. Device number in a READ or WRITE statement is not an integer.
50. An instruction does not end with carriage return and line feed.

#### Compile time queries

- 100
- 101 Variable name has more than six characters.
- 102
- 103
- 104
- 105 A CONTINUE statement is unnumbered.
- 106 Statement immediately following GOTO is unnumbered.
- 107
- 108 A FORMAT statement is unnumbered.
- 109 An EQUIVALENCE statement appears.
- 110 An executable statement occurs before a DIMENSION or COMMON statement.
- 111
- 112
- 113 More than one GLOBAL statement in a program or subprogram.

#### Assembler errors

- EU unlocated label.
- E3 label used twice.
- E5 storage full.

#### Runtime errors

- E1 The number of parameters in the call of the subprogram is not equal to the number in its definition.
- E2 The type of parameters in the call of a subprogram is not the same as the type of its definition.
- E3 Array subscript out of range.
- E4 Computed GO TO variable is out of range.
- E6 Parity error on input.
- E7 Illegal character input in a number.
- E8 First character of an input string is not ` `.

E10 Increment in a DO statement, M3 <= 0  
E11 Attempt to output a non-standardized real number.  
E12 Compiler overwritten.  
E13 Program incorrectly compiled.  
E14 Object code and systems incompatible.  
E15 Real (floating point) overflow.  
E16 Illegal instruction interpreted by QF.  
E17 Overflow on conversion from real to integer value.  
E18 Error in ALOG, EXP or SQRT.  
QF! Incorrect parameter name.  
ROF! Floating point number is too large (overflow).  
RTI! Overflow on real to integer.  
LN! Argument of ALOG, X <= 0.  
SQ! Argument of SQRT, X < 0.  
EX! Argument of EXP, X > 2<sup>16</sup>.



## BASIC

Trigger at 8

Break: interrupt at level 3

INPUT

— delete character  
£ or \ delete line

RUN [line]

CONTINUE (continue after interrupt)

SAVE [line [, line]] (save to paper tape)

LIST [line [, line]] (list to teletype)

DELETE [line [, line]]

NEW (input new program from teletype)

OLD (read new program from paper tape)

KEY (read further program from teletype)

TAPE (read further program from paper tape)

line (delete this line)

line DATA constant,...

line DEF FN letter (simple, ...) = expression

line DIM letter (integer [, integer]), ...

line FOR simple = expression TO expression [STEP expression]

line NEXT simple

line END

line IF expression < expression ...

line IF expression > expression AND ... statement

line IF expression <> expression OR ... statement

line REM ...

line [LET] variable = expression

line THEN|GOTO line

line ON iexpression THEN|GOTO line, ...

line GOSUB line

line RETURN

line PRINT ..., ...

line PRINT ...; ... (; implies no newline after output)

TAB iexpression

"...."

Expression

line READ|INPUT variable, ...

line RESTORE

line SETDIGITS iexpression

line STOP

simple ::= letter | letter digit

variable ::= letter | letter digit | letter (iexpression [,iexpression])

Functions: ABS, ATN, CLG, COS, EXP, INT, LOG, RND, SGN, SIN, SQR, TAN

Operators: + - \* / \*\* ^

Relations: < <= =< = > >= => <> ><



## SYMBOLIC INPUT ROUTINE

### ACD 2-PASS SIR

- 8 Read first tape, first pass, reports in 920 telecode.
- 9 Read subsequent tape, first pass.
- 10 Read first tape, second pass.
- 11 Read subsequent tape, second pass.
- 12 Read first tape, first pass, reports in 900 telecode.  
(You can switch input telecode after any halt code).
- 13 As 10 but ignore first pass errors.
- 14 As 8 or 12 but retain directory.
- 15 As 8 or 12 but retain directory & literals.
- 16 Dump assembler and dictionary.

### Errors

- E0: Instruction Error.
- E1: Contextual Error.
- E2: Octal or Alphanumeric error.
- E3: Label used Twice.
- E4: Global identifier not beginning with a letter.
- E5: Store Full or Patch Error.
- E6: Number Overflow.
- E7: Buffer Overflow.
- E8: Illegal character.
- E9: Stop Code not first character on line.
- EG: Global Error.
- EL: Literal Error.
- EP: Patch Error.
- EU: Unlocated Identifier.

### Options

- 1 list labels - decimal addresses
  - 2 list labels - octal addresses
  - 4 punch zeros for skips
  - 8 set store pointer to 8192
  - 16 set store pointer to 8
  - 32 tie off present binary tape with sumcheck, punch 360 blanks and start new binary tape
- Any multiple of 8192: punch a clear-store for that much memory

### 903 SIR (ACD 1-PASS SIR)

- 8 Cancel all dictionaries and begin assembly of SIR tape.  
(&17740 900 telecode, &17741 920 telecode)
- 9 Assemble SIR tape, maintaining current dictionaries. (&17742)
- 10 Cancel current dictionary, clear store and read RLB tape.  
(&17734 mode 3, &17735 mode 1)
- 11 Read a RLB tape, maintaining current dictionary. (&17736)
- 12 As for 10, but CPA not reset so tape will be loaded following the previous program. (&17737)

### Options

- 1 Display labels
- 2 Load and Go
- 4 Clear store
- 8 Punch loader tape
- 16 Continue assembly at 32
- 32 Set dictionary below program
- 64 Check without assembly



## MASIR & LOADER

MASIR trigger at 16

Ox0 normal assembly - generate RLB  
Ox1 check context - no RLB  
Ox2 macro process only  
Ox3 macro processing with context check  
Ox4 output loader halt sequence, then normal assembly  
Olx preserve macro dictionary from previous run  
  
O options  
M assemble program unit  
C continue after halt

NB rewind tape backwards

LOADER trigger at 16

Oxx1 subsequent tape (vs. first tape)  
Oxx2 library scan (vs. load all)  
Oxx4 produce sum checked binary (vs. load into store)  
Ox1x output binary to paper tape (vs. to backing store)  
Ox2x do not use built-in names (vs. use built-in names)  
Ox4x freeze current dictionary and store layout  
O1xx list labels  
O2xx suppress FIRST LAST messages  
O4xx continue after warnings

N.B. Ox2x required unless names have been built in.

O options  
L load tape  
M program complete. Use nnnnnnM to run if in store, where nnnnnn is octal number to load into A register on entry.

## MAPLOD

Self triggers.

L list located labels and addresses  
U list unlocated labels  
R list references to unlocated labels



## 905 FORTRAN

Trigger at 16

Ox1 syntax check only  
Ox2 data map required (output to punch)  
Ox4 output data map to punch (vs. teletype)  
O1x Use compressed integer array storage allocation

Oxx options  
M compile (or R)

### DATA MAP

YYYYYY aaaa t  
YYYYYY aaaa t bbbbbb  
YYYYYY - symbolic name  
aaaa - relative address  
t - type  
0: undefined  
1: COMMON  
3: local data  
4: indirect address in local data  
bbbbbb - COMMON block name

### LABELS

nnnnnn aaa  
nnnnnn - statement number  
aaaa \_ relative address

### EXTERNAL

pppppp  
bbbbbb ssss  
pppppp - procedure name  
bbbbbb - COMMON block name  
ssss - size

### COMPILER Errors

ttt nnnn llll:  
ttt - type code  
nnnn - last statement number encountered  
llll - count of non-blank lines since this statement number  
ARD array declaratory error  
ASS ASSIGN statement syntax error  
CHx character x expected  
CMN name usage in COMMON  
CN1 illegal constant formation  
CN2 invalid complex constant  
DA1 DATA statement syntax error  
DA2 DATA statement list error  
DO1 DO statement error  
DO2 DO llops not nested  
EQS illegal subscript within EQUIVALENCE  
EX1 replacement operator usage

EX2 unmatched parenthesis within parameter expression  
 EX3 relational usage  
 EX4 illegal operator or operand/operator combination  
 EX5 array or function usage  
 EX6 Incorrect unary usage  
 EX7 illegal type association  
 EX8 Too many open parenthesis  
 EX9 too many closed parenthesis  
 FOR FORMAT statement unnumbered  
 FUN FUNCTION without arguments  
 GTO GOTO statement error  
 HOL Hollerith constant error  
 LIF logical IF error  
 NU1 name missing  
 NU2 variable missing  
 NU3 invalid procedure name  
 NU4 integer variable expected, not found  
 NU5 integer variable or constant expected, not found  
 NU6 array name error  
 PBN procedure name = block name  
 RET RETURN statement in main program  
 RW1 READ/WRITE format reference error  
 RW2 input/output list name error  
 RW3 input/output list syntax error  
 SBX subscript expression syntax error  
 SN1 invalid statement number definition  
 SN2 invalid statement number reference  
 SSQ statement sequence error  
 STF statement function name error  
 STM improper termination of statement  
 STY statement type error  
 TYS type statement syntax error

#### WARNINGS

WD1 unterminated DO loop  
 WD2 illegal DO termination  
 WEX exponent underflow in REAL constant  
 WF1 improper zero in FORMAT  
 WF2 parenthesis nested too deep  
 WF3 improper scale factor  
 WF4 scale factor not followed by conversion format  
 WF5 decimal point missing from conversion format  
 WF6 no digit following decimal point  
 WHC Hollerith constant count error  
 WN1 doubly defined statement number  
 WN2 statement number usage error  
 WN3 no path to this statement  
 WN4 numbered END statement  
 WQ1 Formal parameter of multiple COMMON in EQUIVALENCE  
 WQ2 COMMON base extended back by EQUIVALENCE  
 WQ3 special EQUIVALENCE rules contravened  
 WS2 logical constant spelling  
 WX2 procedure call parameters disagreement

#### MACHINE CODE ERRORS

X1 illegal first character  
X2 function code exceeds 31  
X3 invalid operator  
X4 invalid character  
X5 field too long  
  
YLO invalid logical operator  
YXM Exponentiation mode error  
YY free format line conversion error  
ZZ compiler workspace full

#### RUN TIME ERRORS

First address = call address, second address = operand address

EC1 overflow on conversion to integer  
EDI integer dividend = -131072  
EDZ attempted division by zero  
EML logarithm of negative or zero argument  
EMM exponentiation of negative real argument by negative real exponent  
EOI integer overflow  
EEO exponent overflow  
ESN square root of negative number  
EZZ exponentiation by zero

#### INPUT-OUTPUT ERRORS

EO1 free format specified with WRITE operation, or logical item in list  
for free format output  
EO2 type disagreement between format and list  
EO3 initial character of format is not left parenthesis  
EO4 illegal character in format  
EO5 format syntax error  
EO6 unmatched parenthesis in format, or parenthesis level greater than 2  
EO7 improper use for device type  
EO8 improper QIO call  
WO1 illegal character in data  
WO2 integer out of range  
WO3 exponent out of range

#### CONTROL ERROR REPORTS

QFP EIF invalid parameter code  
QFP EMS invalid mode setting  
QCG ERR index out of range in computed go to



903 UTILITIES

C4 (compare SCB tape with store)  
8 read tape

COPYTAPE

21 read master tape  
22 check master  
23 produce run out for copy  
24 punch copy and check

EDIT

0 read correction tape  
1 read first input tape  
2 read subsequent input tape  
3 re-entry after parity errors  
4 re-entry after stop instructions  
5 read output tape to check punching

FL find line	IL insert on newline
DL delete up to line	IB insert block (^ terminates)
FC find characters	RE remainder of tape
DC delete characters	SH stop on halt
FE find end of line	IH insert halt
DE delete up to end of line	ST stop
IS insert string	CO comment

QCOPY (copy long tapes)

21 Copy tape  
22 Check tape

T22-23

8178 Dump all of store (11111111111110010)  
8179 Dump according to steering tape (11111111111110011)

Data: +8  
+64  
+128  
+512  
halt

AJH MUSIC PROGRAM

8 Read music score into store and play. At end read next tape.  
9 Read music score into store and play repeatedly.  
10 Restart play out of currently loaded music.



## 903 APPLICATIONS

### ESP

ERR1: The internal event list is full.  
ERR2: Attempt to extract an event when the internal event list is empty.  
ERR3: Problem with use of the call procedure: specifically,  $hap \leq 0$ ,  $t < 0$  or  $t + time > 131071$ , or  $send [0] < 0$  or  $> 10$ .  
ERR4: A, N, L or W values mismatch in histogram parameters to hst add or hst sub.  
ERR5: A, N, L or W values mismatch in histogram parameters to hst mult.  
ERR6: Input values for hst in are inconsistent (count of elements in buckets does not match count elements in histogram).

### PERT

32 Read and check control tape  
9 Read data tape

### Errors

1. Checksum failure reading in dump tape
2. Preceding and succeeding events of an activity are the same.
3. No matching activity found in correct or remove mode.
4. Attempt to add more than 511 activities to a project.
5. Attempt to add a duplicate activity.
6. Event number not in range 1-511.
7. Duration larger than 127.
8. Unexpected character.
9. Overall duration read from tape is out of range.
10. Activity earliest start time is greater than 511 time units and/or overall project duration too large.
11. Activity latest start time is greater than 511 time units.
12. Steering commands not in correct work flow sequence.
13. Attempt to input more than 15 commands.
14. Cycle found in project graph.
15. Run number in steering tape wrong.
16. Error in steering tape.



## TEST PROGRAMS

### X.1 INPUT/OUTPUT

- 1 Load using initial instructions
- 2 Jump to 8000 (1111101000000)
- 3 Program punches two blocks of tape
- 4 Load punched tape into reader
- 5 Jump to 8031 (1111101011111)
- 6 Inputs and checks blocks, then punches another

#### Errors:

Punches character read, correct character and stops (7 bits only)

### X.2/X.2B STORE TEST 1 - CHEQUERBOARD

- 1 Load using initial instructions
- 2 Program triggers automatically
- 3 Reads tape after each cycle - blank repeats, anything else halts and 920 STORE TEST 1 O K legible tape punched

#### Errors:

Punches erase, 3 characters, binary address of failed store, 3 characters, incorrect pattern read.

### X.3 FUNCTION TEST

- 1 Load using initial instructions
- 2 Jump to 8 to start
- 3 Reads tape after each cycle - blank repeats, anything else halts and FUNCTION TEST 1 O K legible tape punched

#### Errors:

Punches character - number of failing function (see manual)

### X.4 STORE TEST 2 - ON/OFF

- 1 Load using initial instructions
- 2 Load parameter tape
- 3 Jump to 8000 (1111101000000)
- 4 Reads tape after each cycle - blank repeats, anything else halts and STORE TEST 2 O K legible tape punched
- 5 For on/off test, after above, switch off, switch on, jump to 8009 (1111101001001) and checking should resume
- 6

### X.5 INTERRUPT TEST

- 1 Load using initial instructions
- 2 Set interrupts to MANUAL
- 3 Jump to 8 - responds 4R
- 4 Press level 3 within 5 seconds - responds 3S
- 5 Press level 2 within 5 seconds - responds 2S
- 6 Press level 1 within 5 seconds - responds 1S
- 7 Responds 1T
- 8 Responds 2T
- 9 Responds 3T
- 10 Responds 4R

#### Errors:

See manual

## X.8 TEST TRACE FACILITY

- 1 Load using initial instructions
- 2 Place balnk tape loop in reader
- 3 Jump to 8, tape reads at 20 c/s
- 4 Raise LEVEL ONE interrupt key to TRACE - reading slows down to 1 c/s

Errors: punches one character indicating function digits of last instruction obeyed of the traced program.

## X.10 VARIABLE SPEED PAPER TAPE INPUT/OUTPUT TEST

- 1 Load using initial instructions
- 2 Jump to 8
- 3 Reads in specimen tape and punches two copies
- 4 Load copies into reader
- 5 Jump to 32
- 6 Cycles reading and punching

Errors: halts with incorrect character in A and correct character in Q

## X.33 TELEPRINTER OFFLINE TEST

## X.50 TELEPRINTER TEST

- 1 Load using initial instructions
- 2 Jump to 21 to output test pattern
- 3 Jump to 22 for keyboard test (see manual for keying sequence)
- 4 Jump to 23 for inline test (should echo line after ; or 70 chs)
- 5 Jump to 24 for echo test
- 6 Jump to 25 for reader/punch test

## X.51 PAPER TAPE STATION TEST PROGRAM

- 1 Load using initial instructions
- 2 Jump to 21 - punches a test tape
- 3 Load test tape into reader
- 4 Jump to 22
- 5 Reads in and checks blocks of tape at varying speeds

Errors: punches correct character expected followed by incorrect character read.

## X.52 GRAPH PLOTTER TEST PROGRAM

- 1 Load using initial instructions
- 2 Jump to 21 or 22 or ... or 28 to produce test pattern at different sizes.

## X.202 STORE TEST 1 - CHEQUERBOARD FOR INTERNAL AND EXTERNAL STORES

- 1 Load using initial instructions
- 2 Load parameter tape
- 3 Jump to 7884 (1111011001100)
- 4 Reads parameter tape and checks selected module (0, 1 character for modules, blank to repeat last test. Two non-blanks to terminate.
- 5 Punches STORE TEST 1 O.K. UNIT N

Errors:

Punches erase, 3 characters - failing address, 3 characters, contents.

## X.402 STORE TEST 2 - ON/OFF FOR INTERNAL AND EXTERNAL STORES

- 1 Load using initial instructions
- 2 Load parameter tape
- 3 Jump to 7944 (1111100001011)
- 4 Reads parameter tape and checks selected module (0, 1 character for modules, blank to repeat last test. Two non-blanks to terminate.

5 PUNCHES STORE TEST 2 O.K. UNIT N  
6 Switch off, switch on, jump to 8028 (1111101011100) to resume

Errors:

PUNCHES ERASE, 3 CHARACTERS - FAILING ADDRESS, 3 CHARACTERS, CONTENTS.

XINIT - INITIAL INSTRUCTIONS TEST

1 Load using initial instructions  
2 Trigger at 21, runs phases 1, 2, 3 - I.I. PROTECTED,  
I.I. OVERWRITTEN, PROGRAM OBEYED OVER I.I., ENTER PHASE 4  
3 Trigger at 22 to run phase 4 - ENTER PHASE 5  
4 Trigger at 23 to run phase 5 - END XINIT

QPLOTTEST - TJF PLOTTER TEST

1 Load using initial instructions  
2 Trigger at 8.



TELECODES

920 TELECODE

	00	10	20	30	40	50	60	70	80	90	100	110	120
0 nul	,	4	-	H	R					⑩	d	n	x
1	£	5	.	I	S					<	e	o	y
2 lf	:	6	;	J	T					>	f	p	z
3 pt	&	7	A	K	U					↑	g	q	
4 tab	*	8	B	L	V	sp				~	h	r	
5 bsp	/	9	C	M	W					%	i	s	
6	0	Ⓜ	D	N	X		hlt			?	j	t	
7	1	Ⓜ	E	O	Y					a	k	u	
8 (	2	=	F	P	Z				[	b	l	v	_
9 )	3	+	G	Q					]	c	m	w	del

903 TELECODE

	00	10	20	30	40	50	60	70	80	90	100	110	120
0 nul	lf			(	2	<	F	P	Z	d	n	x	
1				)	3	=	G	Q	[	e	o	y	
2			sp	*	4	>	H	R	£	f	p	z	
3			!	+	5	⑩	I	S	]	g	q		
4			"	,	6	'	J	T	↑	h	r		
5			½	-	7	A	K	U	←	I	s		
6			\$	.	8	B	L	V	@	j	t		
7 bel			%	/	9	C	M	W	a	k	u	del	
8			&	0	:	D	N	X	b	l	v		
9 tab			'	1	;	E	O	Y	c	m	w		

900 TELECODE

	00	10	20	30	40	50	60	70	80	90	100	110	120
0 nul	lf			(	2	<	F	P	Z	d	n	x	
1	pt			)	3	=	G	Q	[	e	o	y	
2			sp	*	4	>	H	R	\	f	p	z	
3			!	+	5	?	I	S	]	g	q	{	
4			"	,	6	@	J	T	^	h	r		
5			£	-	7	A	K	U	~	I	s	}	
6			\$	.	8	B	L	V	˘	j	t	~	
7 bel			%	/	9	C	M	W	a	k	u	del	
8			&	0	:	D	N	X	b	l	v		
9 tab			'	1	;	E	O	Y	c	m	w		

ACD TELECODE

	00	10	20	30	40	50	60	70	80	90	100	110	120
0	nul	lf		(	2	<	F	P	Z	d	n	x	
1				)	3	=	G	Q	[	e	o	y	
2		sp		*	4	>	H	R	£	f	p	z	
3		!		+	5	?	I	S	]	g	q		
4		"		,	6	@	J	T	^	h	r		
5		£		-	7	A	K	U	¯	I	s		
6		\$		.	8	B	L	V	`	j	t		
7	bel	%		/	9	C	M	W	a	k	u	del	
8		&		0	:	D	N	X	b	l	v		
9	tab	'		1	;	E	O	Y	c	m	w		

SIR INTERNAL CODE (903 TELECODE VERSION)

	00	10	20	30	40	50	60
0	sp	*	4	>	H	R	£
1	lf	+	5	⑩	I	S	]
2	"	,	6	'	J	T	↑
3	£	-	7	A	K	U	←
4	\$	.	8	B	L	V	
5	%	/	9	C	M	W	
6	&	0	:	D	N	X	
7	'	1	;	E	O	Y	
8	(	2	<	F	P	Z	
9	)	3	=	G	Q	[	

—