ELLIOTT 900 SERIES SIMULATOR

FORTRAN COMPILATION ERRORS.

Errors are printed out in the form

```
               E n + N
Where     E is the error number, see below,
          n is the last statement number,
          N is the number of statements since n
```

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 subprogram.
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.

Queries raised by the compiler.

These queries occur when a particular statement does not contain a FORTRAN error, but when its effect in 903 FORTRAN may not be quite what the programmer intended. Queries are printed out in the form:

(Q   n + N)

Where Q is the query number, listed below, n and N as for errors.

Q	Cause	Action taken by Compiler
100		
101	Variable name has more than six characters.	First 6 taken as significant.
102		
103		
104		
105	A CONTINUE statement is unnumbered.	Statement is compiled normally.
106	Statement immediately	Statement is compiled normally.

	following GOTO is unnumbered.	
107		
108	A FORMAT statement is unnumbered.	Statement is compiled normally.
109	An EQUIVALENCE statement appears.	Statement is ignored.
110	An executable statement occurs before a DIMENSION or COMMON statement.	The statement is compiled normally unless it contains a variable mentioned in the later specification statement when it will be compiled wrongly.
111		
112		
113	More than one GLOBAL statement in a program or subprogram.	Statement is compiled normally.

Errors detected by the assembler.

EU Messages (unlocated labels).

(i) Qn where n is the statement number referred to by one of the following statements but not subsequently used as a statement number: GO TO, IF, READ, WRITE.

(ii) An EU message giving a subprogram name indicates that the subprogram has not been loaded. A message giving the main program name is ignored.

(iii) A variable has been misspelt in one of the following positions: the right hand size of an arithmetic statement; a WRITE list.

(iv) A variable has not been assigned a value by the program or has been misspelt in one of the following positions: the left hand side of an arithmetic statement; a READ list.

E3 Messages (label used twice).

The label (first six characters only significant) has been used as the name of at least two of the following in the same subprogram: an array; a variable; a subprogram; a library function.

E5 Messages (storage full).

> If running the program in batch mode, rerun in relocatable mode. If already in relocatable mode extra space may be gained by the use of COMMON statements, if appropriate.

Other error messages may be caused by a mispunching in the object tape.

FORTRAN RUN_TIME ERRORS.

Run-time errors are shown on a newline. If continuation is possible, it is made by entry at 9. Error indications take the form:

En A n1 n2 n3

Where n is the error number listed below,
A
N1, n2 and n3 are integers giving further information on the particular error, see below:

E1 A P1 P2
The number of parameters in the call of the subprogram is not equal to the number in its definition.
P1 = number in definition
P2 = number in call
No continuation is possible.

E2 A Q1 Q2
The type of parameters in the call of a subprogram is not the same as the type of its definition.
Q1 = real/integer display in definition
Q2 = real integer display in call
Q1 and Q2 represent a bit pattern indicating which parameters are real:
bit 18 = 1 if parameter 1 is real, = 0 if parameter 1 is integer
bit 17 = 1 if parameter 2 is real, = 0 if parameter 2 is integer
No continuation is possible.

E3 A 1 S1 S2
Array subscript out of range where:
S1 = actual value of subscript
S2 = maximum value allowed
On continuation: S1 is taken equal to 1 if S1 < 1 or to S2 if S1 > S2.

E4 A 1 S1 S2
Computed GO TO variable is out of range where:
S1 = actual value of variable
S2 = maximum value allowed for
On continuation: as for E3.

E6 A C

Parity error on input
C = decimal value of character read
On continuation character is ignored.

E7 A C

Illegal character input in a number
C = decimal value of character
On continuation character is treated as terminating the number.

E8 A C

First character of an input string is not ‵ (acute)
C = decimal value of character
On continuation, character is ignored.

E10 A M3

Increment in a DO statement, M3 <= 0
On continuation, M3 is taken as +1

E11 A N

Attempt to output a non-standardized real number to output device N.
On continuation the number is ignored.

E12 0

Compiler overwritten.
Operator must reload compiler.

E13 0

Program incorrectly compiled.
The operator must re-compile the program, taking care to perform all operations, particularly the entry at 10 to indicate the program is complete.

E14 0

Object code and systems incompatible.
A program has be translated by one issue or version of FORTRAN and attempted to run on an incompatible version.

E15 X A

Real (floating point) overflow.
Ignore X. A is the address in object code where the error occurred.

E16 X A

Illegal instruction interpreted by QF.

(This means the program has been corrupted in some way).
X and A have the meaning of E15.

E17 A

Overflow on conversion from real to integer value.
(Outside the range -131072 to +131071).

E18 A X

Error in ALOG, EXP or SQRT

X = 78 Argument of ALOG(X), X<0
On continuation result = 0.

X = 81 Argument of SQRT(X), X<0
On continuation result = SQRT(ABS(X))

X = 88 Argument of EXP(X), X>2^16
(Also may be caused by exponentiation Y**X).
On continuation, result of EXP(X) is X.

Number too large for style of printing.

If the FORMAT for K is I4 and the value for K at output time is 13,504, the output will take the form **** and continuation occurs automatically.

Errors detected during running.

QF!	Incorrect parameter name.	No continuation possible.
ROF!	Floating point number is too Large (overflow).	Result is taken as largest number of correct sign that can be held.
RTI!	Overflow on real to integer conversion.	Result taken as largest integer of correct sign that can be held.
LN!	Argument of ALOG, X <= 0.	Result taken = +0.
SQ!	Argument of SQRT, X < 0.	Result taken equal to SQRT(ABS(X)).
EX!	Argument of EXP, X > 2^16.	Result taken = X.