

## ELLIOTT 900 SERIES SIMULATOR

### THE 903 FORTRAN SYSTEM.

(For the original Elliott FORTRAN documentation visit: <https://tinyurl.com/Elliott-900-FORTRAN>. Note that not all the features documented are provided in the web-based emulation, in particular there is no support for working in batch mode. If you wish to explore these facilities download my Windows emulator from here: <https://github.com/andrewjherbert/Elliott-900-simulator>).

What kind of FORTRAN is Elliott 903 FORTRAN? Basically it is FORTRAN II with FORTRAN IV style I/O. A FORTRAN program is translated into SIR and then run as a machine code program in conjunction with the FORTRAN runtime routines. Input is free format, and mixed type working is allowed, with automatic type conversion. When real number is converted to an integer it is rounded towards zero.

(The information in this section is derived from Volume 2 of the Elliott 900 Technical Manual.)

### Summary of 903 FORTRAN statements.

<u>Control</u>	<u>Specification</u>	<u>Real functions</u>
IF (E) n1, n2, n3	GLOBAL P1, P2, ..., PN]	ALOG (X)
GOTO n	DIMENSION A(10), K(20,20)	SIN (X)
GOTO (n1, n2, ..., nx) M	COMMON A, K, M, ..., R	COS (X)
DO n M = L1, L2, L3	FUNCTION	ATAN (X)
DO n M = L1, L2	SUBROUTINE	ABS (X)
CONTINUE	FORMAT	SQRT (X)
		EXP (X)
CALL	<u>Input-output</u>	<u>Integer functions</u>
RETURN	READ (L, n) M, R, A, K, ...	IABS (I)
PAUSE	READ (L) M, R, A, K, ...	<u>SIR Code</u>
STOP	WRITE (L, n) Z1, Z2, A, K	CODE
END	WRITE (L) Z1, Z2, A, K	FORTRAN

X represents a real expression

R represents a real variable

E represents any expression

## ELLIOTT 900 SERIES SIMULATOR

I represents an integer expression  
M represents a simple integer variable  
L represents an integer constant or variable  
N represents a statement number  
A is a real array  
Z represents any expression not containing a function.

### Field descriptions. (r=repeat count)

Floating point	rFw.d or Fw.d	(d characters after point)
	rEw.d or Ew.d	(exponent printed)
Integer	rIw or Iw	(w = total print positions)
Alphanumeric	nHabcdef	(n characters)
Spaces	nX	(n spaces)

### Character set.

Elliott 903 FORTRAN uses a subset of printable ASCII.  
Lowercase letters are not permitted, nor are the characters {, \_ or }. In a Hollerith conversion, on output, ! is interpreted as newline.

### Miscellaneous.

A 903 FORTRAN program is preceded by a GLOBAL statement which comprises the name of the program, followed by a list of the subprograms it uses. This list must include the names of any supplied functions, e.g., SQRT, which are used in the program or subprograms. The statement is compulsory for programs but may be omitted for any subprogram which does not itself use subprograms. The statement has the form:

GLOBAL PROG, SUB1, SUB2, SQRT, ATAN ]

Elliott FORTRAN layout is free format: columns are not significant. Statements must be separated from statement numbers by at least two blanks.

Comment lines start with C followed by at least two blanks and then the comment itself. (Lines just consisting of C, with or without blanks, are not permitted).

Continuation lines for GLOBAL and FORMAT statements start with the symbol \ and must be followed by at least two blanks.

## ELLIOTT 900 SERIES SIMULATOR

Programs tapes need to be terminated with a "halt code" (ASCII 20). This can be input to the simulator using a special string "<! HALT !>" on a line by itself. This line can then be followed by data if required. Generally, data should be designed to be self-terminating. If a halt code is encountered in input data the emulation will terminate at that point. (On the real machine, FORTRAN would be waiting for you to load a further data tape.)

Integers can be in the range -131072 to +131071. Integer variables occupy one word of store.

The largest real is about  $9^{18}$ . Real variables occupy two consecutive words of store.

Subscripts of the form -I and -k\*I are permitted as well as the conventional I, k, I+k, I-k, k\*I, k\*I+1 and k\*I-1 forms.

Two dimensional arrays are stored as successive one-dimensional arrays, end to end.

The device number for paper tape input is 4 and for paper tape (punch) output 5. The teletype is device 3.

If an unformatted WRITE statement is used, all the items will be output in format E12.5.

Input is free-format whether or not a READ statement references a FORMAT statement. Numbers must be separated by spaces, tabs and/or newlines. If a real number is input to an integer variable, the value is truncated towards zero.

If a READ statement does reference a FORMAT statement, the FORMAT statement must contain sufficient numeric conversions to read in the desired variables. The mode and field widths in numeric conversions have no effect. Layout conversions (Z, X, /) have no effect. H conversions cause a string to be read and truncated if necessary to the specified length. The string must be surrounded by quotes i.e., ' and @ [900 telecode] or ` (acute) and ' (grave) [903 telecode] (or their alternatives as described in the section on character codes). The quotes are not counted as part of the input string. The opening quote must be the next character to be read by the READ statement as, unlike the input of numeric values, there is no skipping over spaces, tabs and newlines to find the input value.

## ELLIOTT 900 SERIES SIMULATOR

### Restrictions.

1. Integer constants must lie within the range -131071 to +131071.
2. Real constants must lie within the approximate range about  $-9^{18}$  to  $+9^{18}$
3. Any name beginning with Q must have U as its next character.
4. An array cannot exceed 8192 elements.
5. Arithmetic statement functions are not supported.
6. A subprogram cannot have more than 18 parameters.
7. EQUIVALENCE statements are not available.
8. PAUSE and STOP have no arguments.
9. Implied DO statements in READ or WRITE lists are not supported.
10. The READ statement is scanned in its entirety before any actual input of data, therefore code such as  
    J = 3  
    READ (1, 1) J, B(J)  
will assign the second input value to B(3) no matter what value is input to J.
11. If a FORMAT statement is exhausted while values still remain in the WRITE list, the FORMAT statement is repeated from the repeat count, if any, preceding its last left bracket.
12. Repeat counts of a FORMAT sequence (...) are not supported.
13. Device numbers and format statement numbers in READ and WRITE statements can only be integer constants: variables or expressions are not supported.