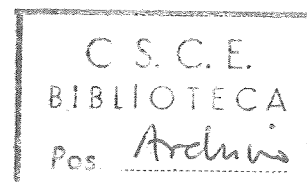


1964

seconda serie



*Pubblicazioni del*

# CENTRO STUDI CALCOLATRICI ELETTRONICHE

*del C. N. R.*

*presso l'Università degli studi di Pisa*

n. 23

# Characteristics of the FORTRAN CEP Language

O. G. MANCINO

*Centro Studi Calcolatrici Elettroniche, C.N.R., Pisa, Italy*

The FORTRAN CEP languages differs from FORTRAN II mainly because: (1) it extends the variety of the modes for real quantities; (2) it allows suitable mixtures, in an input/output list or in an expression, of quantities that occur under different modes; (3) it makes it possible to address a greater number of input/output equipment; and (4) it removes the restrictions on the complexity of the list of quantities to be transmitted between the magnetic core memory and the drum or the magnetic tape units.

The FORTRAN CEP language (defined in [1]) differs from FORTRAN II [2] for the following features.

A quantity may be either Boolean or in any real mode. Generally the mode of one or more quantities is specified by a *mode indicator*. This may appear either on the left or on the right of the opening character of a statement, and is composed respectively, either of a single letter or of a single letter and a colon. There are available without special provisions mode indicators that specify quantities as Boolean, or floating-point single precision, or floating point double precision, or fixed-point (noninteger) single precision, or fixed-point (noninteger) double precision. For real quantities other mode indicators may be used provided that they are introduced by a special statement called *MODE*, but they will be effective only if a suitable group of pseudoinstructions [3] is built to perform the operations required in the indicated modes. The integer quantities are, as in [2], specified by their forms or names; therefore mode indicators have no effect on them.

An expression may contain suitable mixtures of quantities that occur under different modes. Stating it in detail for arithmetic expressions, call *total expression* an expression not contained in another and *argument expression* an expression argument of a function. The value of an arithmetic expression, total or argument, may be obtained in any real mode, called *mode of the total or of the argument arithmetic expression*. Generally this mode must be specified with the suitable mode indicator immediately preceding the total or the argument arithmetic expression. If  $\alpha$  is a total or an argument arithmetic expression immediately preceded by a mode indicator, those noninteger constants, variables and functions, that are contained in  $\alpha$  but not in an argument expression inside  $\alpha$ , are thought of in the mode specified by that indicator. Mode indicators are not necessary before or inside a total arithmetic expression, when it is valid in [2] and it is intended as in [2]. Moreover, it is not necessary to indicate the mode of a total arithmetic expression  $\alpha_1$  when the constants, vari-

ables and functions, contained in  $\alpha_1$  but not in an argument expression inside  $\alpha_1$ , are intended as in [2]. Finally, it is not necessary to indicate the mode of an argument arithmetic expression  $\alpha_2$  when the noninteger constants, variables and functions, contained in  $\alpha_2$  but not in an argument expression inside  $\alpha_2$ , are thought of in the mode of the function of which  $\alpha_2$  is an argument. Integer quantities may appear everywhere in any arithmetic expression.

Each actual argument of a library or of an arithmetic statement function may be either a Boolean expression or an arithmetic expression in any real mode.

Each actual argument of a FORTRAN function or of a CALL statement may be: a Boolean expression, an arithmetic expression in any real mode, an array name, the name of a library function without the terminal *F*, the name of a FORTRAN function, the name of a SUBROUTINE subprogram or a Hollerith field.

An arithmetic statement function may be either arithmetic in any real mode or Boolean.

The list of quantities to be transmitted may be, in any input/output statement with which it is associated, as described in [2, pp. 37-38]. In the list several mode indicators may appear, each of which may be introduced before any list element and also before any variable which is not part of a subscript or of indexing information. A mode indicator specifies the mode of the noninteger variables that follow it, that is, until another mode indicator occurs. For variables of the list which are not preceded by a mode indicator, the mode is as in [2]. Extraparentheses are not required.

Several readers and punches may be addressed. and reading of paper tapes may be done with or without the control of a FORMAT statement.

Output on a typewriter is allowed and is caused by a suitable statement called *TYPE*.

A FORMAT statement may include up to three levels of parentheses.

In a DIMENSION statement several mode indicators may appear and may be introduced before any array name. A mode indicator specifies the mode of noninteger arrays, referred to with names following it, until another mode indicator occurs. For matrices whose names are not preceded by a mode indicator, the mode is as in [2].

*Acknowledgments.* Thanks are expressed to Dr. A. Caracciolo di Forino and Dr. I. Galligani who contributed valuable ideas for the FORTRAN CEP language.

RECEIVED JANUARY, 1964; REVISED MARCH, 1964

## REFERENCES

1. MANCINO, O. G. FORTRAN CEP language, a FORTRAN II version for the CEP. Centro Studi Calcolatrici Elettroniche, Pisa, Italy, 1963.
2. IBM Reference Manual 704 FORTRAN Programming System. Form C28-6106.
3. Manuale delle istruzioni CEP. Centro Studi Calcolatrici Elettroniche, Pisa, Italy, 1960.