CC-186-11

COMPUTATION CENTER
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

TO: FMS Users

FROM: Judith Spall

DATE: October, 1964

SUBJECT: FORTRAN and MAD Format Specifications

The following memo is a description of the format specifications used in MAD and FORTRAN codes programs. The material was taken largely from the MAD Manual in hopes of giving a more detailed explanation of input-output for MAD and FORTRAN than is now available.

PRELIMINARY DEFINITIONS

These definitions are complete in themselves, meaning and usage of them will be explained in the course of the write-up.

1. Character

   A character is one of the 64 combinations of two octal digits (00 to 77) in the IBM 7094. When printed on an IBM 1403 printer, they print as 49 distinct "characters" where blank is considered to be a character also.

2. Digit

   A digit is one of the following characters

   1 2 3 4 5 6 7 8 9 0

3. Letter

   A letter is one of the following characters

   A B C D E F G H I J K L M N Ø P Q R S T U V W X Y Z

4. Special Character

   A special character is one of the following characters

   = ' , ($ * .) / + - ≠

5. Number

   A number is a string of one or more digits.

6. List

   A list is a sequence of addresses telling the locations of storage cells from which information should be taken for output or into which information should be put from input.

7. Format

   A format is a string of characters (packed 6 to the word) which tells the form and positioning of the input or output.

8. Format term

   A format term is a string of characters between two break characters (normally a "," or "/"). It does not include the break characters. This may be an empty string. (I.e. Consecutive break characters are allowed).

9. Format terminator

The format terminator is a character which signifies the end of a format. In formats for MAD calls this is an asterisk in normal context. In formats for FØRTRAN calls this is the right parenthesis in normal context which matches the left parenthesis that was the first character of the format.

10. Control character

A control character is either a letter or a special character. It is the character which designates what the format term it is in will do. Each format term has exactly one control character. e.g. The control character "A" designates BCD conversion.

11. Modifying character

A modifying character is either a letter or a special character. It cannot stand alone in a format term. It must always appear with a control character. Its presence changes or modifies the action of the control character. e.g., The modifier "R" with the control character "C" causes characters to be read in to be right-justified instead of left-justified. Since a format term is not processed until the break character that separates it from the following one is hit, a modifying character can occur anywhere in the term, except that modifiers which require constants (B and P) must come before the multiplicity.

12. Field

A field is a set of contiguous columns on an input card or record or an output card, line or record.

13. Field width

The field width is the number of columns in the specified field. If the field width is zero in a format term, for input a zero will be read in, for output the number will be put in zero columns. i.e., The list item will be skipped.

FORMAT SPECIFICATIONS

When information is read from (or punched in) a card into (or from) a computer, it is necessary to know how this information has been allocated among the available columns of the card. Similarly, whenever information is to be printed by a printer (either on-line or off-line), it is necessary to know how this information has been allocated among the columns available on the printer. A description of each allocation is called a format specification. Usually, but not always, such a specification is accompanied by a list of variables whose values are to be printed or whose values are to be read. (Occasionally, the information is contained entirely in the format specification, so the list may be empty.) In FORTRAN a format specification is a string of characters (as described below), enclosed in parentheses and in MAD, is a string of characters, enclosed in parentheses or terminated by an *. These specifications are stored, six characters per computer word, (BCD Mode) in arrays. These arrays may be preset, computed, or read in as data.

AVAILABLE COLUMNS FOR PRINTING, PUNCHING, AND READING

Every format specification consists of a description of the allocation of available columns, and the form in which the particular information is to appear. Specifications for reading or punching cards and printing of information are identical, with the following exceptions:

a. CARDS — A card has only 72 available columns, if read through the on-line card reader, or if punched through the on-line card punch and has only 80 available columns, if processed through off-line equipment.

b. PRINTING — A line of print has 120 available columns if printed on-line or off-line with any off-line equipment other than an IBM 1401. Off-line printing done on the IBM 1401 allows 132 available columns. As will be explained under CARRIAGE CONTROL, the first column of print plays a special role.

THE BASIC FIELD DESCRIPTION

Each specification describes successive fields across the available columns, starting from the left. If the specification describes fewer than the total number of available columns, the line (or card) will automatically be filled in with blanks.

The basic field description consists of a letter followed by an integer. The letter indicates the form of the information in the external medium as follows:

I	Integer
F	Fixed-point number (internally floating-point number)
E	Floating-point number
Ø	Octal number
K	Octal number (for MAD only)
A	Alphanumeric information
C	Alphanumeric information (for MAD only)
G	Fixed- or floating-point numbers

(In MAD, for the purpose of input/output, the Boolean values of 0B, 1B are considered as fixed-point integers and will be punched in cards and appear in print as 0, 1.)

The integer indicates the size of the field; i.e., the number of available columns to be used. For example, Ø3 indicates a three-column octal field, A23 indicates a 23-column alphanumeric field.

For E, F, and G fields, i.e., fixed- and floating-point numbers, there is the question of the placement of the decimal point, the form of the numbers, etc., in addition to the field size.

For this reason, the basic field description for E, F, and G fields require an additional integer giving the number of places after the decimal point that are to be rounded and printed (or read, or punched).

Thus, the basic field description consists of the following:

1. Type (i.e., I, F, etc.)
2. Width of field expressed as an integer
3. A decimal point.
4. Number of decimal places to be printed expressed as an integer.
5. A comma, unless the description is at the end of the format. The comma serves as a separator for each description.

It should be noted that 3 and 4 apply to E, F, and G conversion only.

THE FORM OF THE NUMBER

Integers (I fields) and octal numbers (Ø, K fields) are printed, punched, etc., directly, without any decimal point. Numbers printed or punched in E fields have the form (if 5 decimal places are requested, for example):

$$\pm.X_1X_2X_3X_4X_5E\pm N_1N_2$$

where N N is the exponent of 10 by which the number in front of the E is to be multiplied. On input cards numbers in E fields must have an exponent of the form

$$E\pm N_1N_2$$

Similarly, if the "E" is punched, a "+" may be omitted, as well as a leading zero in the exponent. The exponent must be counted in the field size and it must be right adjusted in the field on input cards.

Numbers to be printed or punched in E fields have the form (if 5 decimal digits are requested, for example):

$$\pm X_1X_2X_3.X_4X_5X_6X_7X_8$$

although "+" signs may be omitted, but not "-" signs. One should note that if no decimal places are requested the decimal point is omitted in the output.

On both E and F input data on cards, blanks are regarded as zeros, and any number of digits may be used, but only 8 digits of accuracy are retained. Moreover, E and F input data need not have the decimal point punched. If the decimal point is not punched, the field specification determines its position. For example, the punched number +9032, right adjusted, described by the specification F10.2 would be understood to be the number +90.32 because the 2 in the specification indicates 2 decimal places to the right of the point. Similarly, the punched number +9032E3 described by the specification E10.4 would be understood to be the number +.9032E3. If the decimal point is punched, it completely overrides the setting of the decimal point by the specification. The entire specification must be present, however.

Within the numeric type fields, E, F, G, I, K, and Ø, "+" signs are not printed or punched nor are they necessary on input cards. However, "-" signs may not be omitted on input and are always printed and punched. If the field description gives a field size larger than the number required, the number is pushed as far right as possible. If the field size given in a specification is too small, a diagnostic will occur, unless the programmer executes the statement CALL IØHSIZ (X) where X is non-zero. In such a case the information is printed, punched, or read from the right end of the field until the field is exhausted, and the left end of the number is truncated, including the sign. It is important, therefore, to be sure to provide a large enough field to handle the information expected. In fact, some spacing can be achieved by giving large field sizes, since blanks automatically occur to the left of a number pushed to the right end of an oversized field.

G CONVERSION

The G conversion is available for printed or punched output. The variable to be printed or punched is scanned to see if the prefix, tag, and address are all zero in which case the variable is assumed to be a FORTRAN type fixed-point integer and the I conversion is used. If the variable's prefix, tag, or address is not zero, the location is assumed to contain a floating-point quantity. If the quantity is less than 10^{-d} or greater than 10^{d}, if d (the number of decimal places desired) is less than 10, the E conversion is used, otherwise, the F conversion is used.

For example, if a programmer wished to print out the following variables I, J, A, and B according to the format

(4G13.5)

and the contents of their locations were as follows in octal:

Prefix	Decrement	Tag	Address	
0	00001	0	00000	I
0	00012	0	00000	J
2	01400	0	00000	A
1	75400	0	00000	B

the printed output would like like this:

1 10 1.00000 ±.62500E-01

The contents of land J are FORTRAN-type fixed-point variables stored in the decrement. The remainder of the machine word is zero. Consequently, the I conversion is used with a field width of 13. The variables A and B are assumed to be floating-point variables since either the prefix, tag, or address is not zero. Since $10^{-d} \leq A \leq 10^{d}$, where d = 5, the F conversion is used, since B does not lie in that region, the E conversion is used.

For input fields, G is identical to F.

PERMISSIBLE CONVERSION

It should be understood that there must be a relationship between the form of a number inside the computer and its external form. In other words, a number described by an E or F specification is assumed to be in floating-point form in storage. Information described by a G field must be in fixed-point or floating-point form in storage. Similarly, a number governed by an Ø (K in MAD) specification will be handled by the direct binary octal conversion. Information described by an A (C in MAD) specification is assumed to be in alphanumeric (BCD) form both inside the computer and outside. Note that if FORTRAN users wished to use the K or C conversion it would be necessary to read in the format and terminate it with an *.

REPETITION OF BASIC FIELD SPECIFICATION

If several consecutive fields can be described by the same basic specification, repetition may be avoided by prefixing the basic specification by its multiplicity. For example, the specification (3F10.3, E18.4, 2E9.1, 3I2) is a short way of writing (F10.3, F10.3, F10.3, E18.4, E9.1, E9.1, I2, I2, I2). Either specification may be used of course. A group of basic specifications may be repeated by enclosing the group in parentheses and preceding the left parenthesis by the multiplicity. Thus, (3E10.3, 2(I2, 3F10.1), 2A6, 2G18.9) would be the equivalent of the following:

```
(E10.3, E10.3, E10.3, I2, F10.1, F10.1, F10.1, I2,
    F10.1, F10.1, F10.1, A6, A6, G18.9, G18.9)
```

Note that in FORTRAN type formats such groupings of parentheses may not be nested and if the format is exhausted before the list, the format will now begin with the number preceding the last left parentheses. (See page 13).

SCALE FACTORS

One extra feature is allowed when reading, printing or punching E and F fields. A scale factor may be applied to an F number according to the formula:

$$\text{External number} = \text{Internal number} * 10^{\text{Scale Factor}}$$

(where the scaling is accomplished before the conversion is done and the scale must be between -8 and +8). Note: On input this means internal number = external number * $10^{-\text{Scale Factor}}$. The "scale factor", followed by the letter "P", is prefixed to the whole field specification, until a new scale factor is given as in the example:

```
(-2P2F7.3, F7.3, 0PF7.3)
```

Thus, four numbers which would print 0.522, -1.567, 93.671, 93.671 according to the specification (4F7.3) would print instead:

.005 -.016 .937 93.671

if the specification (-2P2F7.3, F7.3, 0PF7.3) were used. It must be noted that for F fields this scale factor actually changes the values of the numbers to which it applies. For E fields, the "scale factor" causes the number itself to be modified so the true value of the number remains unchanged. Thus, the number 0.9321E-3 would print as .9321E-03 according to the specification E18.4, but it would print as 93.2100E-05 according to the specification 2PE18.4. Unlike an F number, the value is the same in either case. In MAD programs, where the format is not enclosed in parentheses, but terminated by an *, the scale factor "P" does not refer to the whole format. For example, the four numbers 0.522, -1.567, 93.671 and 93.671 would print according to the format -2P2F7.3, 2F7.3*:

.005 -.016 93.671 93.671

MULTIPLE SPECIFICATIONS

Several specifications may be condensed into one larger one by the use of the character "/". Each appearance of a "/" (except as part of a Hollerith field) indicates that the specifications of a new line (or card), is to be started with what follows. One should beware of using slashes at the beginning and end of format statements. (See page 13). A pair "//" causes a blank line (or card), three "///" causes two blank lines (or cards), etc. Thus the specification

(3F10.3/I2,Ø18,2A5/7I3)

implies that one line (or card) is described by the specification 3F10.3, and the next line (or card) is described by the specification I2, Ø18, 2A5, and the next by 7I3. It must be noted that each line (if printing is being described) is described individually here, and must include its own carriage control. (Explained on page 11.)

ALPHANUMERIC FIELDS

It has been stated on the previous page that a number appearing in an over-sized field is pushed as far as possible to the right. In the case of A information, if there are more than siz characters, the six right-most characters are transmitted as input and are right adjusted in an output field. In the case of C, in MAD, if there are more than six (6) characters, the six left most characters are transmitted as input and are left adjusted in an output field. Similarly, in case the specification describes a field less than six (6), characters are taken from the left end of the field until the field is exhausted, and stored in the left end of the machine word followed by blanks.

Thus, if a card contains the characters: ABCDEFGHIJK in column 1 through 11, and it is read according to the specification 2A3 (or 2C3 which is permissible in MAD), the two six character words are read into the computer.

ABC (with three trailing blanks
DEF on each)

while the specification A6 would cause a single word to be read: ABCDEF and A7, A3 would cause the words:

BCDEFG and HIJ (with three trailing blanks)

to be read (since at most, six characters can go into one word of storage).

HOLLERITH FIELDS (LABELS)

Although the A (C in MAD) specification is available for transmitting characters (Hollerith information) to and from storage, it is often convenient to include strings of Hollerith characters as labels or heading directly in the format specification. This is done by means of a basic Hollerith field specification consisting of the strings of characters to be transmitted, preceded by a count of these characters and the letter "H". Thus, if the specification (1H1,F10.3,6HBETA = E10.2) were used in printing, one would obtain a new page, because of the one column Hollerith field, containing the character "1". Then a ten column F number would print, followed by the characters "BETA =" and a floating-point ten column field. Note that blanks are completely ignored throughout all format specifications except when they occur as characters in a Hollerith string. Note also that while every field specification of types G, I, K, C, S, T, E, F, Ø, and A must be followed by commas, the comma may be omitted after a Hollerith string of the type described here. The comma may be used, however, if desired.

Hollerith strings appearing in a format specification used during input operations are replaced by the information appearing in the next columns. If the specification (9H F6.2) were used to read in the data card 1THETA = 10.32 the Hollerith information 1THETA = would now be within the format specification in the machine as follows: (9H1THETA = F6.2).

If information were to be printed or punched according to this specification once such a data card were read, the Hollerith information which was read would appear in the new output rather than the Hollerith information which was compiled with the problem. For example, if printing was to appear, the output at the top of a new page would be THETA = 100.32.

In all MAD programs and in those FORTRAN programs, where the format statement is read in as data and terminated by an *, a Hollerith string may be specified without the count preceding the H. This may be done by having the first character following the H taken as "a break character", and all characters in the format between it and the next appearance of it are used as the Hollerith field. (See the example on page 14.)

WARNING: There is danger here of writing several such Hollerith strings without counts and exceeding the maximum number of columns allowed for a line (or card). If the number of columns used approaches the maximum, the counts should be used.

BLANK FIELDS

The format specification "X" is used to indicate a column to be skipped or ignored while reading data cards, or to be left blank if printing or punching. An integer must precede the X to indicate repetition even if the number is one. The comma may be omitted after the X specification.

In MAD the format specification "S" may be used to indicate a column to be skipped or ignored. An integer must follow the S to indicate repetition even if the number is one. The comma may not be omitted after the S specification. Note that FORTRAN users may use the S specification provided the format containing it is read into the computer.

For example, if one wished to read an integer in column 10, skipping columns 1-9, the format specification might be written (9XI1). If the format were written (I10), the information in columns 1-9 would not have been ignored.

CARRIAGE CONTROL

The first column of a card has no special significance. The first (left most) character of a line of print is treated differently. This character governs printed carriage control, such as skipping to a new page, double spacing, etc., and should not contain information to be printed. The user has effectively 131 available characters on a line of print with an IBM 1401 for off-line processing (otherwise 119 available columns), but he must always include the first character as a code to control the vertical spacing of that line. For example, the specification (16X, 6HBETA = I2) will indicate that the line (or card) starts with a skip of 16 columns and then prints (or reads or punches) the characters "BETA =" followed by an integer field of two columns. If a card is involved, the letter "B" in "BETA" will be found in column 17, due to the skip of 16 columns. However, if a line is to be printed, the "B" will print in column 16 of the printed page, since the first character of the line (in this case, the first of the 16 columns to be skipped) is detached, to be interpreted as a carriage control leaving an effective skip of 15 columns. The effect of a "blank" character on carriage control is to move to the next line before printing, as will be seen from the following table. In each case, the carriage control character is effective before the rest of the specification is carried out unless otherwise indicated.

Blank	Single Space
0	Double Space
+	No Space (Note that overprinting does take place on the M.I.T. 1401)
-	Triple Space (Effective only on the IBM 1401)
/	Single Space, but do not go to the top of a new page after 60 lines, but after 66 lines. (Effective only on the M.I.T. 1401)
S	Double Space, but do not go to the top of a new page after 60 lines, but after 66 lines. (Effective only on the M.I.T. 1401)
T	Triple Space, but do not go to the top of a new page after 60 lines, but after 66 lines. (Effective only on the M.I.T. 1401)
1	Skip to next page.
2	Skip to next half-page.
3	Skip to next third-page.
4	Skip to next quarter-page.

As another example, the specification (6H1PHI =F6.3) would cause a skip to the next page (because the first character is a "1"), and cause "PHI =" to be printed, followed by a fixed-point number. As will be explained below, the "6H" that appears in the specification indicates that 6 Hollerith (or BCD) characters follow, e.g., "1PHI =". Note that blanks are counted as characters here. It should be understood that this special behavior with regard to column 1 pertains to lines that are to be printed, (either on-line or off-line), and not at all to cards.

RELATIONSHIP BETWEEN THE LIST AND THE SPECIFICATION

The "list" consists of a set of names of variables to or from which information is to flow. Except for Hollerith strings imbedded in the format specification itself and the X and S fields, each field in the specification refers to one item such as A(6)...A(8), counts as several names of variables (in this case, the three variables A(6), A(7), and A(8)). During the transmission of information, the input or output subroutine scans both the list and the specifications simultaneously, correlating corre-- ponding entries, and associating a field size, a type of conver- sion, etc., to each variable. If a Hollerith string is encountered in the specification, it is immediately transmitted, and it is not associated with any item of the "list".

For example, if the list consisted of: A, B(1,1), I, K where I and K were integers, and the others floating point, and the specification was (1H1 F11.2, E14.4, 3X 3HI =I3, 9X3HK =I3) we might find a printed line something like the following (at the top of the next page because of the 1H1):

```
456.010  -.1613E+04      I = 50          K = 17
```

The same list would look the following way with the specification (1H1 2F11.3, 6X 3HI =I3, 9X3HK =I3):

```
456.010  -1612.510       I = 50          K = 17
```

As stated above, a specification may not account for more than 80 columns on a card. It may happen, however, that a list calls for more information than can appear on a single card, or perhaps only a certain part of each card is to be read. The determining factors in every case is whether or not one may account for the entire list. After each card is read according to the format specification, the list is consulted, if not yet satisfied, another card is read, and so on. It is important to realize that the specification is not necessarily scanned from the beginning when a new card is read. In fact, the specification scanner moves to the left from the end of the specification (the last right parenthesis) until it hits a left parenthesis not in an H field. (If there is no left parenthesis, it will move to the the beginning of the specification.) It then examines the characters just to the left of this left parenthesis to see if they are a multiplicity indication. From this left parenthesis (together with the multiplicity if any) to the end of the specification now becomes the format specification until the list is satisfied. A similar statement may be made for printed or punched output.

Thus, in the specification: (3F10.3/4(F10.2, 6HBETA = I2)) the first line printed (or read, or punched) would have three fixed- point numbers and subsequent lines would all be printed (or read, or punched) according to the specification 4(F10.2, 6HBETA = I2).

As an example, one might have an integer on the first data card, followed by many cards, each with six floating-point numbers. The specification might then be (I6/(6E10.5)). Only the first six columns would be read on the first card, and only 60 columns would be read on subsequent cards. The remaining columns are ignored and may contain any legitimate Hollerith characters. Note that if one wished to read one fixed point number according to this same format specification, a data would be skipped due to the /.

If a specification contains a Hollerith string of the form $n\ H\ a_1a_2\ldots\ a_n$, certain conventions are observed. (1) If the list is satisfied, but the next field specification is a Hollerith string, the string is transmitted anyway. (2) On the input, i.e., reading from cards, when a Hollerith string is encountered in the specification, the information in the corresponding columns of the input card will be brought in and will replace the BCD string itself within the format specification. This can then be used as a specification for output, for example. This is useful for labeling a set of data and causing the label to appear on the output along with the data, etc.

Thus, a card punched as follows:

```
1 DATA SET NO 3-A   JULY 31, 1964   J. DOE
```

might be read with the specification:

```
(72H                                                              )
```

Later, this specification could be used to print the same information as a heading for the results. Note the "1" provided for carriage control for the printing.

In all MAD programs and in FORTRAN programs where the format is read on a data card it is possible to use a T field specification to refer to a different column in the line (or card) image being processed; e.g. T35, 4HABCD* would cause the characters ABCD to be put into columns 35-38.

As an example, the specification

```
T20,H*AND/OR*,I5*
```

will cause the characters AND/OR to appear followed by a five column integer.

WARNING

The specification 72X and 72 (1H) while indicating 72 blank spaces, does not allow the reading into the computer of an entire card, as indicated above, since they do not really provide a storage region of 72 characters in length into which the information on the card, may be read and stored until needed.
program where the format statement is read in on data cards

MODIFIERS

Modifiers are only available in all MAD programs where the format specification terminates with an * and in any FORTRAN program where the format statement is read in on data cards and terminated by an *. It has been shown that field specifications have the standard form nqw.d, where n is the multiplicity, q is the control letter, w is the field width, and d is the number of characters after the decimal point (if present). Several modifiers may be prefixed to this field specification in MAD programs to cause special effects. Some modifiers (D,L,R,Z) do not require integers immediately in front of them while others (B, and P previously described) do. The order in which modifiers occur is immaterial. Examples of modified specifications are:

-3P4F19.6, LK6

The modifiers have the following effect:

B - Normally, conversion is performed from (to) a decimal external form to (from) a binary internal form. The B modifier allows the use of other external number systems for I conversions. Thus, the modifier 16B causes the external form to be in hexadecimal (base 16) notation. The base thus specified may not exceed 19. For integers greater than 10, the additional characters needed are taken from the beginning of the alphabet. (For base 16, (a = 10, b = 11, c = 12, d = 13, e = 14, and f = 15.) Note that K12 is not exactly equal to 8BI12, since the K12 conversion uses the left-most bit as part of the number, while the 8BI12 uses this bit as the sign of the integer. The numbers occurring in the format specification itself are always interpreted as being in the decimal system.

D - Double Precision - if the character D precedes an E or F specification, it indicates that this conversion is to be performed on a double-precision number (with each half of the number containing its own characteristic and fractional parts). Both parts of the number must be specified on the "list" the high order half first. In FORTRAN one would print an array of 5 elements as follows:

```
PRINT FMT, (A(I), A(I+5), I=1,5)
```

L - This is used with numbers brought into the computer with a K specification. They are usually right-adjusted, with leading zeros inserted. The L modifier causes the number to be left-adjusted (i.e. shifted left to eliminate all leading zeros), with trailing zeros supplied.

M - Floating Dollar Sign Modifier - When this modifier is used in E, F, G, or I <u>output</u> format terms, a dollar sign is inserted into the field immediately to the left of the first digit. If the number is negative (so there is a minus sign), the dollar sign goes immediately to the left of the minus sign.

N - "Don't" Modifier - If the N modifier occurs before a slash in an output format, it means don't blank out the line and don't reset the line pointer in column 1 after printing.

If the N modifier occurs before the format terminator in an output format, it means don't print the line or reset it when exiting (if the list is exhausted).

These two applications of "N" allow building up of a print line column by column, adding each number with a separate I/Ø statement, but not printing the line until it is complete. This avoids using the "+" carriage control and wasting printer time.

R - This is used with characters brought in by a C specification. They are usually left-adjusted, with trailing blanks inserted. The R modifier causes the characters to be right-adjusted (i.e. shifted right to eliminate trailing blanks), with leading blanks supplied.

V - Commas Every Three Digits Modifer - When this modifier is included in an E, F, G, or I output format term, it causes commas to be placed every three digits to the left of the decimal point (or to the left of the right end of the number, if there is no decimal point). These commas must be allowed in the specification of the field width.

W - Blank If Zero Modifier - If the modifier W occurs in an A, C, E, F, G, I, K, or Ø output format term, any number that is all zero will not be printed out. Instead, blanks will be put in its place. Note that this does not mean that zeros cannot be printed out - a number may very possibly be not all zero and yet when converted, round to zero.

Z - This forces leading or trailing blanks to be replaced by zeros on a C specification.

USE OF FORMAT VARIABLES (in MAD coded programs only)

A format variable may be used at any point in a format specification at which an integer constant is normally found; i.e., within any field specification or a part of any modifier. Each time a format is interpreted (i.e., an input/output statement referring to it is executed), the current value of each format variable encountered is used in the format in place of the name of the format variable. Primes are used to delimit format variables when used in format information. Format variable occurence may have one of three possible forms:

'V', 'V(I)', or 'V(I_1,I_2)'

where V is a format variable and I, I_1, and I_2 are either integer constants or format variables. Also format variable V, I, I_1 and I_2 may be of either floating point, integer or Boolean mode. Boolean and floating point values will be converted to integers before being used. 1B and 0B will be converted to 1 and 0, respectively and floating point values will be truncated as usual. Thus, in the format 'A'F3.1, F10.6*, where A is a Boolean format variable, the specification F3.1 will be used if A has the value 1B, and not otherwise. All format variables must be names that exist in the program in which the input/output statement using the format occurs (i.e., they may not be arguments to a function which uses them as format variables), and they must be declared to be format variables using a FORMAT VARIABLE declaration.

EXAMPLES:

(1) Values of X(I) can be plotted verses time (as represented by lines in the paper) as follows (assuming column 66 to represent 0):

```
      THRØUGH QQ, FØR I=1,1,I.G.N
QQ    PRINT FØRMAT ØUT
      VECTØR VALUES ØUT = $T66,S'X(I)',IH**$
      FØRMAT VARIABLE X,I
```

(2) The following formats are legitimate (although highly improbable):

```
'SCALE'P'NBR'F'WIDTH'.'DECDIG'*
'A(1)'P'A(2)'E'A(3)'.'A(4),'A(5)'B'A(6)'I'A(7)'*
```

(3) Usage as switches:

```
H= THE TRIANGLE IS=, 'SWITCH'(H*NOT*)
H= A RIGHT TRIANGLE= *
```

and assuming SW1=.NOT.SW2, the following might be used:

```
'SW1'(5E)'SW2(5F)15.5
```

Note that the multiplicity for the E and F must be inside the parentheses to keep it separate from the parenthesis multiplicity represented by SW1 or SW2.

WARNING:

When using a format variable for the multiplicity, remember that varying the multiplicity does not vary the number of items on the list. If it is necessary to skip list items, use data-transmission format terms with zero field width to do it.