

PAUL:

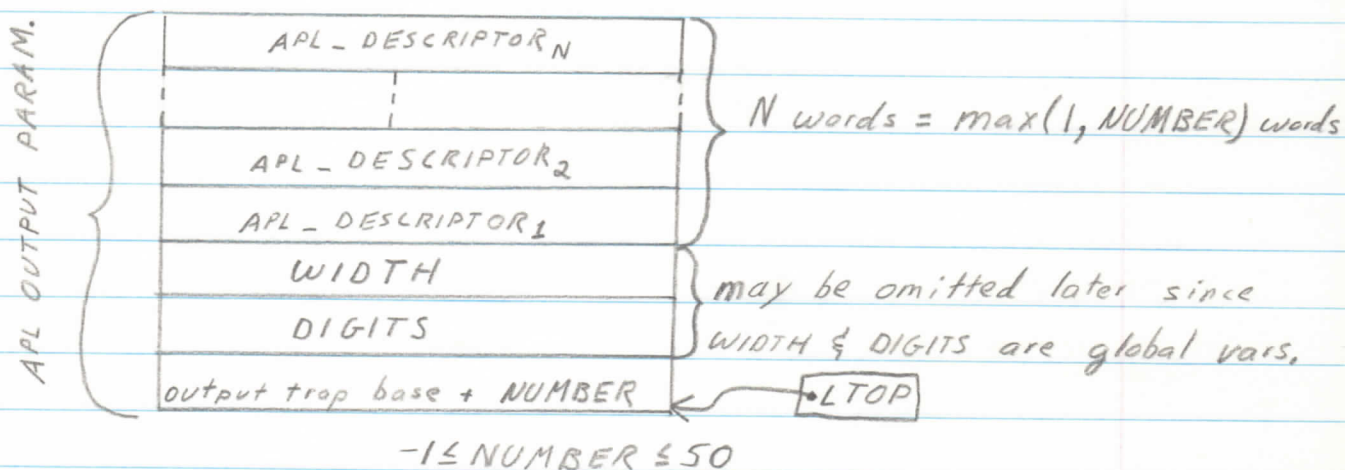
APL OUTPUT CONTROLLER

tw6

I Parameters

A. pass by value by APL program

The APL stack looks like this after the APL OUTPUT Attn-trap:



B. pass by reference by the Line Collector

SIMPLE scalar H = the horizontal position of the terminal wheel. The APL OUTPUT CONTROLLER does not change H , but may reference H before and after calling Line Collector functions, such as NEWLINE(), that may change H .

C. the APL stack after servicing the output trap

Since

$$\begin{cases} \text{NUMBER} > 0 \Rightarrow \text{explicit output} \\ = 0 \Rightarrow \text{quad output} \\ = 1 \Rightarrow \text{quote-quad output} \end{cases}$$

agreed
implied

the stack eating algorithm is:

$$\begin{cases} \text{delete all APL OUTPUT parameters if } NUMBER > 0 \\ \text{leave only APL-DESCRIPTOR}_1 \text{ otherwise} \end{cases}$$

The reference counts of deleted APL-DESCRIPTOR_i arrays must be decremented. (If the count becomes zero, the array space should be returned to the free pool.)

II Logic Diagram

A. explicit output: $NUMBER > 0$

```

FUNCTION EXPLOUT()
    QUOTEQUAD(APL-DESCRIPTOR1)
    ⋮
    QUOTEQUAD(APL-DESCRIPTORN-1)
    QUAD (APL-DESCRIPTORN)
    RETURN
END

```

B. quad output: $NUMBER = 0$

```

FUNCTION QUAD(DESC)
    IF QCOMMON(DESC) THEN NEWLINE()
    RETURN; END

```

C. quote-quad output: $NUMBER = -1$

```

FUNCTION QUOTEQUAD(DESC)
    QCOMMON(DESC)
    RETURN; END

```

D. common logic

1. parameter check

IF RANK(DESC)=1 AND NELTS(DESC)=0 THEN RETURN TRUE

IF DESC#TYPE=UNDEFTYPE OR NELTS(DESC)=0 THEN RETURN FALSE

IF WIDTH or DIGITS invalid THEN FRETURN

When done, RETURN TRUE

2. initialize H if necessary

IF (H>1 AND RANK(DESC)>1) OR WIDTH<H..
THEN NEWLINE()

3. conversion rules

-SCALARS: treat like one element vectors

-VECTORS: exactly two blanks separate numbers from numbers; no blanks separate characters from numbers, numbers from characters, and characters from characters; no blanks precede the first element nor follow the last element of the vector; numeric data representation is accurate to DIGITS significant digits; all trailing zeros past the decimal point are suppressed, as is the decimal point itself, whenever possible.

Use E-notation for all numbers:

$$-1,0001 < X < 1,0001$$

$$X > 10^D$$

$$X < 10^{-D}$$

1) less than $1E^{-4}$ but greater than $-1E^{-4}$, or

2) not less than $1E<DIGITS>$, or

3) not more than $-1E<DIGITS>$

Use F-notation for all other numbers. Numbers between -1 & 1 exclusively should have a zero to the left of the decimal point, in F-notation.

- ARRAY of rank > 1 : numeric elements of array are printed according to a numeric field specification. Numeric fields always start with one or more blanks, and end with zero or more contiguous blanks. No blanks separate numeric fields, from numeric fields, characters from numeric fields, and characters from characters.

The E-notation numeric field specification takes up w print positions, where

$$w = 7 + \text{DIGITS} - \begin{cases} 1 & \text{if DIGITS} = 1 \\ 0 & \text{otherwise} \end{cases}$$

if, for example, $\text{DIGITS} = 4$, field = 'bSX.XXXEnnn'

Note: E-notation may have trailing zeros past decimal point.

Use E-notation if:

- 1) any numeric element satisfies 1), 2), or 3) above.
- 2) $\text{DIGITS} < 1 + \max_i \{ \text{EXP}_i \} - \min_i \{ \text{EXP}_i - \text{SIGNIFDIG}_i \}$

where $i \in$ all numeric array elements

Otherwise, use Fw.d ($d=0 \Rightarrow$ don't print dec. point)

$$\text{where } d = \max \{ 0, -\min \{ \text{EXP}_i - \text{SIGNIFDIG}_i \} \}$$

$$w = 3 + \max \{ 1, \max \{ \text{EXP}_i \} \} + \begin{cases} 0 & \text{if } d=0 \\ d+1 & \text{if } d>0 \end{cases}$$

if, for example, $w=6$, $d=3$, field = 'nn.fff'

Numbers between -1 & 1 should have a zero to the left of the decimal point.

4. printing rules

- always keep $H \leq \text{WIDTH}$
- wrapped around lines are indented 6 spaces
- never break a line in the middle of a number
- if an array dimension is $d_1 d_2 \dots d_{n-1} d_n$,
insert a CRLF after every d_n elements, &
insert a LF after every $d_{n-1} \times d_n$ elements, &
insert another LF after every $d_{n-2} \times d_{n-1} \times d_n$ elements, &
0
0
0