コアダンプは16進で印字されるが，ハードとしては整数型で16ビット（＝1語），実数型で前述のように32ビットを使用しての2進である。が注意すべきことは，この2進表示においては文字，数字の区別をしていないということである。

以下その例をあげてみよう。

下記のようなプログラム（EX 1）があったとき

```
      ＊＊＊SOURCE  LIST＊＊＊
ISN          STATEMENT
  1  C       ＊＊＊＊ EX 1 ＊＊＊＊
  2          DATA   L/2 H   ＊/
  3          WRITE (6,600)  L，L
  4     600  FORMAT (1H0,20X,I5,5X
             ,A2)
  5          STOP
  6          END

          16476          ＊
```

WRITE 文の記述子を「A 2」とするか「I 5」にするかによって，2進表示から意味のはき出しが変ってくる。

ダンプしてみると

4 0 5 C

がデータとして「L」に格納されていることがわかる。これを「A 2」で記述させてみると

␣＊

なのだが，これを（4 0 5 C）を2進に直してから10進数で表現させてみると

0100 0000 0101 1100

$1\times2^{13}+1\times2^{6}+1\times2^{4}+1\times2^{3}+1\times2^{2}$

$=16476$

という意味になっているので，これを「I 5」で記述を命令させてみたから

1 6 4 7 6

を印字するのである。

このように，内部では常に文字，数字の区別はなく，同じ2進のままなのである。

記述子によってプリンタへの結果印字があらわれることになっているのである。

したがって上例の場合は偶然「␣＊」または，「1 6 4 7 6」がでてくれたけれど，一般的には下の例にみられるような場合が多いので十分このことを留意してかからなければならない。

```
        *** SOURCE LIST ***

ISN          STATEMENT

  1  C       ***** EX3 *****
  2          DATA B/4H    */
  3          IB=B
  4          WRITE(6,600)B,B
  5     600  FORMAT(1H0,20X,A4,5X,E15.7)
  6          WRITE(6,601)IB,IB
  7     601  FORMAT(1H0//21X,A2,7X,I5)
  8          STOP
  9          END
```

ダンプすると「B」には

4 0 4 0　　4 0 5 C

で，これは2進化すると

4　0　4　0　4　0　5　C

0100 0000 0100 0000 0100 0000 0101 1100

すなわち，16進による仮数部は

（0．4 0 4 0 5 C）

であり，指数部は0乗なので16^{0} となるから

$(0．4\,0\,4\,0\,5\,C)_{16}*16^{0}$

$=(4\,0\,4\,0\,5\,C)_{16}*16^{-6}$

が生じてくる。

この式から10進法の計算をしてみると

$(4\,0\,4\,0\,5\,C)_{16}=(4\,2\,1\,0\,7\,5\,2)_{10}$

なので

$$\frac{4\,2\,1\,0\,7\,5\,2}{16^{6}}=(0．25098)_{10}$$

を得る。

よってIタイプで記述すれば「0」である。

そして「Aタイプ」で記述させると，IB内は

IB＝B

によりIB＝0となっているので

$(0\,0\,0\,0)_{16}$

のようなものが格納されていることになる。

よって，表1に該当するものがない場合はすべて