

FORMAT= FREE or BINARY or RB4 or RB8
or DATABANK or EXCEL or LABELS or LOTUS or 'フォーマット文';

機能：

FORMAT は READ と WRITE コマンドとともに用いるオプションで、TSP プログラム内でのデータの読み込みおよび書き出しに使われます。このセクションでは FORMAT 文の作り方の説明を行います。どこで使うかについては、READ と WRITE を参照して下さい。

使用法：

FORMAT にはいくつかの方法があります。

1. FORMAT=FREE では、数値が自由形式で読み込まれます。すなわち、一字以上の空白によって数値は区別されますが、数値を区切る空白の長さが一定である必要はありません。
2. FORMAT=BINARY では、データがバイナリー形式で読み込まれます。その場合、各変数には浮動小数点型単精度の文字が当てはめられます。バイナリー型のデータは、他の形式のデータと混在することはできません。しかもある計算機から、他の型の計算機に移して利用することも出来ません。FORMAT=RB4(実数4バイトバイナリー)は同じです。
3. FORMAT=RB8は、倍精度バイナリー(8バイト)です。
4. FORMAT=DATABANK では、データファイルが TSP のデータバンク機能によって作られた事を示します。この方法は IN あるいは OUT/KEEP といった機能によって置き換えることも出来ます。
5. FORMAT=EXCELは、エクセル表計算ファイルを読み込みます。ファイル名が .XLS で終わる場合はこれを用います。
6. FORMAT=LABELS は WRITE に対してだけ使えます。すなわち、標準的な PRINT コマンドの時のようなラベルを使うことが出来ます。
7. FORMAT=LOTUSは、ロータス123ワークシートファイル(.WK1, .WKS)を、最初の行を列名、そしてもしあれば最初の列を日付として読み込みます。ファイル名に .WK が含まれていれば、このフォーマットを用います。
8. FORMAT=RB4は、FORMAT=BINARY(4バイト)と同じです。
9. FORMAT=RB8は、倍精度バイナリーに用います。
10. FORMAT= 'フォーマット文' の方法では、データが読み込まれる特定の形式(フォーマット)を定めます。TSP における形式は FORTRAN 形式のプロセッサ(フォーマット処理用のプログラム)が TSP で使われているので、FORTRAN の形式と非常によく似たものです。しかし、TSP の利用者は単純なフォーマット文を作成するために FORTRAN のプログラミングに習熟している必要はありません。TSP の操作に必要な形式の定め方は、以下で説明します。

FORTRAN プログラマーへの注意: TSP で使われるデータは総て浮動小数点ですから、整数型と文字型は使わないで下さい。またフォーマットを定める際に、その機能が確実にないかぎり括弧による変数のグループ分けはしないで下さい。なぜなら計算機を動かしている OS により処理が異なり、思わないエラーを生じるからです。

フォーマット文の始まりと終わりは ' ' か " " であり、引用符のすぐ後と前に括弧が続きます。TSP はフォーマット文中に括弧があるかどうかを調べ、もし抜けていれば自動的に括弧を挿入します。従ってフォーマット文のなかで引用符を使うことはできません。ラベル付きの WRITE 文では、FORMAT=LABELS, TITLE コマンド、あるいは H 形式 (HOLLERITH) のフォーマット文を使って下さい。

括弧の中でのフォーマット形式は、データレコードの中での数値の長さや小数点の位置などを述べるものです。TSP プログラムで通常使われるフォーマット形式は、X, F, E, と G です。X は読み込みの際に飛ばされるべきカラム数を定めます。F は浮動小数点に使われます。E は指数形式での浮動小数点を定めます。G はアウトプットに使われて、状況により F か E かもっとも適切な形式を選びます。

次に、フォーマット (F5.2) を使って 5 列の数値を読み込む例をあげましょう。

```
Col: 12345
      10000
```

この例では数値は 100.00 と読まれます。なぜなら、このフォーマット文は 5 桁の数値で小数点の後 2 桁であることを表しています。

次に (F10.5,5X,E10.3,F8.0) というフォーマット文の例を考えましょう。

```
      0      1      2      3      4
Col: 1234567890123456789012345678901234567890123456789
      343.5      .98765E-01      200
```

この例では、先ず最初のフォーマットが 10 桁の内 5 桁は小数点の後にくる事を示しますが、データとして与えられた数値に既に小数点が含まれているので、この数値は 343.5 と読まれます。次の 5X は 11 桁目から 15 桁目までを読み飛ばす事を示しています。

E10.3 は指数型の数値を読みますが、指数型は数値が通常の表示には小さ過ぎるか、あるいは大き過ぎる場合に使われます。また 10 は 16 桁から 25 桁に対応し、3 は小数点以下に 3 桁の数字がある事を示します。しかしデータには既に小数点が含まれているので、読まれた値は .98765 掛ける 10 のマイナス一乗、すなわち .098765 となります。最後のフォーマットは F8.0 ですが、これは 26 から 33 までのカラムに対応し、かつ小数点以下に数字がない事を示しています。この場合は、インプットの数値に小数点がないので、読み込まれた値は 200 となります。

フォーマット文はいろいろな方法で組み合わせて使うことが出来ます。例えばあるフォーマットの前に整数値がついていれば、そのフォーマットが整数回繰り返されることを意味します。いくつか例を示しましょう。

```
(8F10.5)
```

これは 80 カラムのレコードに、10 カラムの数値が 8 個あることを意味しています。

```
(10X,5E12.6,10X,8F3.1)
```

では、104 カラムのレコードに 13 個の数値があり、5 個は E 形式、そして 8 個は F 形式である事を示します。第 1 カラムから第 10 カラム、そして 71 カラムから 80 カラムは読み飛ばされることが分かります。

```
(20F5.2/20F5.2)
```

この例では、観測値当り2レコードでそれぞれが20変数であることを意味しています。(/ は、新しいレコードに移ることを意味します。)

この最後の例は、TSP で入力されるフォーマットはデータの重要な特色を示しています。一般的には、ある観測値は必ずフォーマット文を通して読まれたり書かれたりされます。したがって、フォーマット文は、読みこまれたり書き出したりしたい変数の数に丁度等しいだけのフォーマットがなければいけません。通常1レコードは、各観測値毎に読まれます。それはその観測値に対する全ての変数を含んでいなくてはなりません。しかしスラッシュ(/)を使えば、複数のレコードでもって一観測値とすることが出来ます。レコードは通常は同じ長さですが、必ずしも同一のフォーマットである必要はありません。