

1. FORTRAN の変数名：先頭が英文字で 8 文字程度以下の英数字。大文字と小文字の区別はない。

(例) I X ABC K20 B1X3 (5CX は、先頭が数字なのでダメ)

2. 暗黙の型宣言

- ・型宣言をしていない変数で、変数名の先頭文字が I,J,K,L,M,N のいずれかなら整数型変数、それ以外は(単精度)実数型変数

- ・IMPLICIT 文により、上記の取り決めを変更することができる。

(例) IMPLICIT REAL*8 (A-H,O-Z) 変数名の先頭文字が、A~H と O~Z なら倍精度実数

注) 暗黙の型宣言文については、一部プログラマーから酷評される場合があるが、よく理解して使用すれば、

これほど便利なものはない。暗黙の型宣言文を無効にする命令もあるが、変数すべての型宣言が必要。

3. 型宣言文 整数型 INTEGER 単精度実数型 REAL 倍精度実数型 DOUBLE PRECISION

単精度実数は REAL*4 倍精度実数は REAL*8 でもよい。

文字型 CHARACTER あるいは CHARACTER*文字数

4. 配列変数：同じ変数名で大量のデータを扱うために使用される。DO ループ等で効果的に利用できる。

- ・最初に配列宣言が必要：配列宣言文 (例) DIMENSION ABC(100),IMAX(10,30),NOZOMI(3,10,5)

- ・ABC(100) は、ABC(1)~ABC(100) の 100 個のデータ

- ・IMAX(10,30) は、IMAX(1,1)~IMAX(10,30) の 300 個 (10×30) のデータ

5. FORTRAN (固定形式) のプログラム (以下は、固定形式のものである。それ以外に、自由形式がある。)

- ・基本的に、1 行に 1 文しか書けず、命令文は、7 桁目 (第 7 カラム) から 72 桁目までに書く。

- ・文番号は 1~5 カラム (1 桁目~5 桁目) に書く。なお、6 桁目は継続行の印に使われる。

- ・注釈行 (コメント行) の印 (C あるいは *) は第 1 カラム (1 桁目) に書く。

(行の先頭に C あるいは * を書いた場合、その行は意味を持たない)

- ・1 行 (72 桁目まで) で書ききれない命令文は次の行以降の 7~72 桁目にまたがって書くことができる。

続く行を「継続行」と呼び、継続行の第 6 カラムには「0 (ゼロ) 以外の文字」を入れる (空白記号もダメ)。

継続行は 19 行まで続けられる (最初の行と合わせて 20 行まで)。

- ・行の途中で、以降をコメントにしたい場合には、そこに ! を書いたうえでコメントを書く。

- ・命令文の途中の任意の場所に「空白」を入れることができ、それはコンパイル時に無視される。

(ただし、「空白」そのものに意味がある場合は、その限りではない。)

6. 文番号は、「GOTO 文のジャンプ先」、「繰返し文 (DO 文) の端末行」等に用いる。

7. $A=B+C$ のような命令文を代入文と呼ぶ。右边で計算した結果を左辺の変数に代入する命令。

従って、 $A=A+2$ というような数学的には矛盾するような命令も意味を持つ。

なお、 $A+B=C+D$ という命令は誤りである。

足し算記号 + 引き算記号 - かけ算記号 * わり算記号 /

べき乗記号 ** 平方根 SQRT() 自然対数 LOG() 常用対数 LOG10()

絶対値 ABS() 正弦 SIN() 余弦 COS() 正接 TAN() 逆正弦 ASIN()

- ・整数同士の割算は「切り捨て」となるので注意 (例) $8/3+1$ の計算は 3 となる

- ・優先順序を示すカッコは、丸カッコ()を何重にも用いる (例) $C=(A+B*(X**(K+3)-Y))/(P-5.8)$

8. 入力命令 (並びによる入力の例) READ(*,*) IA,B,XYZ,S20

標準の入力装置 (パソコンではキーボード) から変数の並びの順にデータを入力する。上の例では、

例えば、10,20.3,123.45,56 と入力すると $IA=10$, $B=20.3$, $XYZ=123.45$, $S20=56.0$ となる。

入力データの区切りの印には、「コンマ」あるいは「空白」を用いる

9. 出力命令（出力の例）

① `WRITE(*,*) A,XYZ,I20`

② `WRITE(*,600) ABC,IP`
`600 FORMAT(F10.5,I8)`

③ `WRITE(3,*) K5,BBB,Z80`

- ・ `WRITE` 文の括弧の中の最初の記号は「出力先」を示す（* は標準出力，パソコンではディスプレイ）．次の記号は「書式」を示す（* は並びによる出力）．
- ・ 「出力先」に「ファイル名等」を指定するには，`OPEN` 文，`CLOSE` 文を用いるが，詳細は省略する．
- ・ 通常は，出力を終えた後「改行」するが，下記の書式の最後に \$ を入れると「改行は行わない」

10. 書式の例

整数型書式 `I10` : 文字型書式 `A8` `A`

実数型書式 `F9.4` `E15.7` `G18.9` (例) `123.45` → `123.45 (F6.2)` `0.12345E+03 (E11.5)`

任意文字の出力（ `apostrophe` で囲む） `'Mie University'`

X書式（空白の出力） 例 `5X` , H書式（任意文字の出力） 例 `14HMie University`

11. FORTRAN の命令の実行順序

基本的には，主プログラムの最初に書かれた命令文（宣言文を除く）から順に下に向かって実行される．ただし，途中にジャンプ文（`GOTO` 文）があると，そこからは，指定行へ強制的にジャンプする．また，分岐文があると，条件に応じて実行される命令文が異なってくる．

12. ジャンプ文（指定する行の命令へジャンプする）

`GOTO` 文番号

13. 分岐文（条件判断文，`IF` 文）には，論理 `IF` 文，ブロック `IF` 文，算術 `IF` 文がある（大まかに言って）．

①論理 `IF` 文： `IF(条件式) 命令文` (例) `IF(A.GT.B) GOTO 100`

括弧の中の「条件式」を満たせば「命令文」が実行される．「命令文」を実行した後は，次の行に実行が移る．ただし「命令文」が `GOTO` 文なら，指定された行へジャンプする．なお，条件式を満足しなければ，そのまま次の行に実行が移る．

②条件式としてよく用いるものに以下のようなものがある．

(記号で書く場合) -----

・ 等しい	<code>==</code>	(例)	<code>I2 == K</code>
・ 等しくない	<code>/=</code>		<code>J /= 100</code>
・ 大きい	<code>></code>		<code>P3C > XYZ</code>
・ 大きいか等しい	<code>>=</code>		<code>MC >= N5</code>
・ 小さい	<code><</code>		<code>X < 12.34</code>
・ 小さいか等しい	<code><=</code>		<code>NN <= 300</code>

上記の関係演算子は下記 2 つの論理演算子より優先順位が高いので，下のカッコは付けずともよい．

・ 論理積	<code>.AND.</code>	<code>(I == 5).AND.(A < B)</code>
・ 論理和	<code>.OR.</code>	<code>(I == 5).OR.(A < B)</code>

(文字で書く場合) -----

・ 等しい	<code>.EQ.</code>	(例)	<code>I2.EQ.K</code>
・ 等しくない	<code>.NE.</code>		<code>J.NE.100</code>
・ 大きい	<code>.GT.</code>		<code>P3C.GT.XYZ</code>
・ 大きいか等しい	<code>.GE.</code>		<code>MC.GE.N5</code>
・ 小さい	<code>.LT.</code>		<code>X.LT.12.34</code>

- ・ 小さいか等しい .LE. NN.LE.300
- ・ 論理積 .AND. (I.EQ.5).AND.(A.LT.B)

③ブロック IF 文：

IF(条件式 1) THEN

条件式 1 を満足した場合に実行する命令文（複数行可）

途中に GOTO 文で外へ飛び出さないならば、この後 ENDIF の次へ実行が移る

ELSEIF(条件式 2) THEN

条件式 1 を満足せず、条件式 2 を満足した場合に実行する命令文（複数行可）

途中に GOTO 文で外へ飛び出さないならば、この後 ENDIF の次へ実行が移る

ELSEIF() THEN

.....

.....

ELSE

いずれの条件も満足しない場合に実行する命令文（複数行可）

途中に GOTO 文で外へ飛び出さないならば、この後 ENDIF の次へ実行が移る

ENDIF

(例) IF(N.GE.100) THEN

 A=A+COS(X)

 ELSE

 B=B+SIN(X)

 ENDIF

④算術 IF 文： IF (算術式) 文番号 1, 文番号 2, 文番号 3

- ・ 算術式の結果が「負」なら文番号 1 へ、「ゼロ」なら文番号 2 へ、「正」なら文番号 3 へ
実行が移る（ジャンプする） (例) IF(M+N) 10,20,30

14. 繰り返し文 (DO ループ)：制御変数が最終値を越える手前まで、命令文が繰り返し実行される。

DO 文番号 制御変数=初期値, 最終値, 制御変数に加算される値

繰り返し実行される命令文（複数行可）

文番号 最後の繰り返し実行文（一般的には、CONTINUE 文がよく用いられる）

(例) S=0

S=0

DO 20 I=1,10

左右どちらも

DO 20 I=1,10

DO 10 J=1,30,5

同じ計算内容

DO 20 J=1,30,5

IMAX(I,J)=I*2+J

IMAX(I,J)=I*2+J

10 S=S+XYZ(I,J)

S=S+XYZ(I,J)

20 CONTINUE

20 CONTINUE

- ・ 上の例では、制御変数 J は、1, 6, 11, 16, 21, 26 のように変化して命令が繰り返される
- ・ 制御変数に加算される値が 1 の場合は、1 は省略できる。 (例) DO 20 I=1,10

○繰り返し文の DO ループに文番号を使わない方法もある

DO 制御変数=初期値, 最終値, 制御変数に加算される値

繰り返し実行される命令文（複数行可）

ENDDO

(例) DO I=1,10,2

 DO K=1,30

IMAX(I,K)=.....

XYZ(I,K)=.....

ENDDO

ENDDO

15. CONTINUE 文とは：文番号だけは保持するがそれ以外には特に何も実行しない命令文
16. FORTRAN プログラムは、1つの主プログラムと複数（ゼロでもよい）の副プログラムから成る
17. 主プログラムと副プログラム、あるいは、異なる副プログラムにおいては、変数名は独立して用いられる（同じ変数名が用いられても、引数や共通ブロックの中で名前を一致させない限り、異なる変数となる）
18. 副プログラムは「プログラムを整理して見やすくするため」と「同じような処理を何度も繰返して記述しないで1回の記述で済ますため」等利用される。
19. 副プログラムには、関数副プログラムとサブルーチン副プログラムがある（大まかに言って）。

①関数副プログラム： FUNCTION 関数名（引数の並び）

（例） FUNCTION XG3(A,B)

- ・括弧の中に並んでいる変数を引数と呼び、異なるプログラム間のデータの受け渡しに用いる（呼び出す側の引数を実引数、呼び出される側の引数を仮引数と呼び、実引数と仮引数それぞれで、変数の個数は同じで、変数の並びの順に型が一致している必要がある。このことは、サブルーチン副プログラムでも同様である）
- ・関数副プログラムを呼び出すには、呼び出す側の命令文の中でその関数を使用すればよい

（例） X=Y+XG3(P,Q)

②サブルーチン副プログラム： SUBROUTINE サブルーチン名（引数の並び）

（例） SUBROUTINE SYZ(P,G,R)

- ・サブルーチン副プログラムを呼び出すには CALL 文 を用いる。

（例） CALL SYZ(A,B,C)

20. 主プログラムと副プログラム、あるいは、副プログラム間でデータを受け渡しする方法

①引数（実引数と仮引数）による方法

- ・実引数と仮引数の中の変数の型は、それぞれの並びの順に同じ型（一致した型）でなければならない。

②共通ブロック（コモンブロック）で定義した変数による方法

- ・共通ブロックの宣言 COMMON /共通ブロック名/ 変数の並び

（例） COMMON /B001/ A,XC9,P(10),HH(3,2)

- ・同じ共通ブロック名の「変数の並び」に置かれた変数の型は、並びの順に同じ型（一致した型）でなければならない。

③関数副プログラムの場合は関数名そのものが変数となるので、その変数名による方法

21. 文関数：1行（1つの式）で定義できる関数は、関数副プログラムを使わないで、プログラムの先頭で定義して用いることもできる。これを文関数と呼ぶ。

（例） BBC(X,Y)=SQRT(X*X+Y*Y)

22. 計算実行を停止する文（プログラムの必要な場所に何か所でも入れられる）

STOP 停止箇所を特定するには、 STOP 123 あるいは STOP 'abc'

23. 副プログラムから主プログラムに戻る箇所に書く文（必要な場所に何か所でも書ける）

RETURN

24. プログラム単位の記述の終わりを示す文（プログラム記述の最後に入れる。主プログラムと各副プログラムの終わりにそれぞれ必要）

END