

Title	FÖRTRAN L (500) と700の比較について
Author(s)	後藤, 米子
Citation	大阪大学大型計算機センターニュース. 9 P.31-P.53
Issue Date	1973-02
Text Version	publisher
URL	http://hdl.handle.net/11094/65180
DOI	
rights	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/repo/ouka/all/>

FÖRTRAN L(500) と 700 の比較について

研究開発室 後藤 米子

日本電気より、マニュアル

MÖD III/IV FÖRTRAN L と

MÖD IV EX/VII FÖRTRAN 700 比較説明書

が出されていますが、これはマニュアル上の比較ですので、その点をお含み下さい。

詳細は、上記のマニュアルに譲りますが、以下に、2つの FÖRTRAN の比較のためのいくつかのプログラム例をあげます。

テストに使用した FÖRTRAN コンパイラーは

FÖRTRAN 700 Rev. 030 030

FÖRTRAN L(500) Rev. 055

です。

FÖRTRANL(500)と FÖRTRAN 700 は、以下では略して FL(500), F700 と書きます。

(左 FÖRTRAN L500)
(右 FÖRTRAN 700)

1. データの領域

(説明) FL(500) では、データの領域(DIMENSION や COMMON の宣言で確保できる記憶領域のことは、262Kch までしか許されません。

次の例では $300 \times 120 \times 8 \div 1000 = 288\text{Kch}$ となり、エラー・メッセージが出力されます。なお F700 の場合は、2Mch. をこえた場合に、エラー・メッセージが出力されます。

```
C      TEST PROGRAM DATA AREA
      DIMENSION A(300,120)
      STOP
      END
```

```
SEQ    DIAGNOSTIC MESSAGE      | エラー・メッセージなし
      111 DATA EXCEEDS 262 K OF
      STORAGE                    |
FATAL ERRORS PREVENT SUCCESSFUL |
LINKLOADING AND EXECUTION      |
```

2. 配列の次元

(説明) FL(500)では、3次元までの配列しか許されていませんが、F700では、7次元まで使用できるように拡張されています。

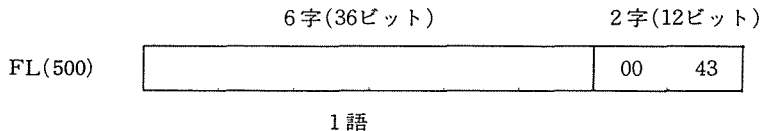
```
DIMENSION AD(1,2,3,4,5,6,7)
C=1.0
C=C+1.0
STOP
END
```

```
SEQ    DIAGNOSTIC MESSAGE
0001 039 MORE THAN 3 DIMENSIONS
      ARE DECLARED FOR ARRAY "AD"
FATAL ERRORS PREVENT SUCCESSFUL
LINKLOADING AND EXECUTION
```

エラー・メッセージなし

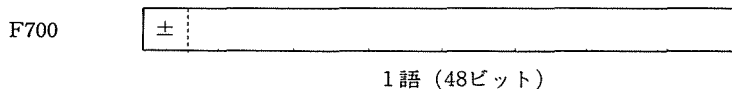
3. 整数型データのとりうる範囲

(説明) FL(500)と、F700の整数型データの内部表現は次の通りです。



48ビットの非正規化2進浮動小数点形式がとられます。

指数部は常に $0043_8 = 35_{10}$ で2の35乗を表わしています。



48ビットの2進補数形式がとられる。

FL(500)では、-34, 359, 738, 367~34, 359, 738, 367 (10進10桁強)

F700では、-140, 737, 488, 355, 328~140, 737, 488, 355, 327

$$(2^{-47} \sim 2^{47} - 1)$$

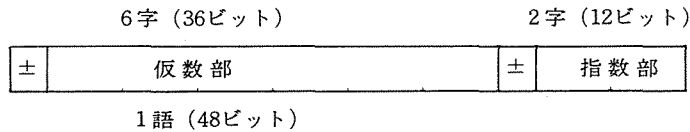
なお、参考のために、オーバー・フローしたときの値がどうなるかを知るために、テストプログラムを添えておきます。

(省略)		(省略)	
4294967296	—4294967296	4294967296	—4294967296
8589934592	—8589934592	8589934592	—8589934592
17179869184	—17179869184	17179869184	—17179869184
0	—34359738368	34359738368	—34359738368
0	—34359738368	68719476736	—68719476736
0	—34359738368	137438953472	—137438953472
0	—34359738368	274877906944	—274877906944
0	—34359738368	549755813888	—549755813888
0	—34359738368	1099511627776	—1099511627776
0	—34359738368	2199023255552	—2199023255552
0	—34359738368	4398046511104	—4398046511104
0	—34359738368	8796093022208	—8796093022208
0	—34359738368	17592186044416	—17592186044416
0	—34359738368	35184372088832	—35184372088832
0	—34359738368	70368744177664	—70368744177664
0	—34359738368	0	—140737488355328
0	—34359738368	0	—140737488355328

4 尾数型データのとりうる範囲

(説明)

F700 では、実数型データの内部表現は次のようになっています。



浮動小数点形式で表現されます。

仮数部は、その右端のビットを 2^{-35} の位とする 2 進形式、

指数部は、その右端のビットを 1 の位とする 2 進形式で格納されます。

仮数部の値が m で、指数部の値が e なら、実数型データとしては、 $m \cdot 2^e$ の大きさを表わします。

なお参考のために、オーバー・フローしたときの値がどうなるかを知るために、テスト・プログラムを添えておきますが、この結果は検討を必要とします。

<pre> 0001 0002 0003 0004 0005 0006 .9574977455-603 </pre>	<pre> PS=1.0E600 DO 21 I=1, 30 21 PS=PS*10.0 WRITE(3, LIST) PS STOP END </pre>	<pre> .9574977455-603 OVF OPTION を指定したとき **400N EXPONENT OVERFLOW ERROR DETECTED IN NFSANCHOR 00007714 CALLED FROM MAINPG 0003 00003450 0 .9574977455-603 </pre>
--	--	--

5. 整数型データのゼロ除算

(説明) テスト・プログラムのように、結果が異なります。F700では、OPTIONOVFを指定することによって、実行時に40/N ZERO DIVIDEのエラー・メッセージが出力されます。

```

0001      I0=0
0002      I=3
0003      I1=I/I0
0004      I2=I/0
0005      WRITE(3,LIST) I0,I1,I2
0006      STOP
0007      END

```

0	3	3		0	0	0
---	---	---	--	---	---	---

6. 実数型データのゼロ除算

(説明) 5.と同じです。

```

0001      AI0=0.0
0002      AI=3.0
0003      AI1=AI/AI0
0004      AI2=AI/0.0
0005      WRITE(3,LIST) AI0,AI1,AI2
0006      STOP
0007      END

```

.0000000000	3.000000000	3.000000000		.0	.0	.0
-------------	-------------	-------------	--	----	----	----

7. 添字式

(説明) F700では、整数型、整数型の変数を+、-、×で結んだものが使用できるように拡張されています。()と/は使用できません。なお、DATA文中では、JIS型のみが許されます。

```

DIMENSION A(10)
DO 16 II=1,10
16 A(II)=FLOAT(II)
I=I
J=1
K=1
B=A(I*J+K+2*K)
C=A(I-K+J+K)
WRITE(3,LIST) B,C
STOP
END

```

```

SEQ  DIAGNOSTIC MESSAGE SUBPROG-RE
      0007 062 SYNTAX ERROR IN SUB- 4.000000000  2.000000000
            SCRIPT OF ARRAY EL-
            MENT REFERENCE TO
            "A"
      0008 062 SYNTAX ERROR IN SUB-
            SCRIPT OF ARRAY ELE-
            MENT REFERENCE TO
            "A"
FATAL ERRORS PREVENT SUCCESSFUL
LINKLOADING AND EXECUTION

```

8. D \bar{O} の制御変数

(説明) D \bar{O} の制御変数も、7.と同様に拡張されています。

```

0001    DIMENSION A(5)
0002    N=1
0003    M=3
0004    DO 10 I=1, M+N+1
0005    10 A(I)=FLOAT(I)
0006    WRITE(3, LIST) A
0007    STOP
0008    END

```

```

SEQ    DIANGOSTIC MESSAGE
0004 016 ILLEGAL DO STATEMENT 1.000000000  2.000000000  3.000000000
      SYNTAX 4.0000000000  5.000000000
0005 029 WARNING: UNREFERENCED
      STATEMENT LABEL
0005 020 WARNING: UNDEFINED VARIABLE
      "I"
0005 020 WARNING: UNDEFINED VARIABLE
      "I"

```

9. D \bar{O} ループをぬけ出したあとの終値パラメーターの値

(説明) JIS F \bar{O} RTRANになりますと「D \bar{O} が満足された場合には、制御変数は不定になる」とありますので、再定義しないと使えません。

```

0001    S=0.0
0002    DO 22 I=1,10
0003    22 S=S+1.0
0004    WRITE(3, LIST) I
0005    STOP
0006    END

```

11

10

10. サブルーチンの引数の型

(説明) 文法に関する補足と、その他の注意の(23)にもあげましたように、F700では、対応する実引数と仮引数の型の一致は検査されていません。実行時には実引数の型の内部表現が仮引数の型の内部表現が仮引数の型の内部表現とみなされて実行されます。テスト・プログラムのメインでNは暗黙の型宣言に従い整数型ですが、サブルーチンのzは実数型ですので、結果が異なっています。

(メイン)

```
0001      N=2
0002      X=2.0
0003      CALL ARG(N, X, Y)
0004      WRITE(3, LIST) N, Y, Y X
0005      STOP
0006      END
```

(サブルーチン)

```
0001      SUBROUTINE ARG(Z, XX, YY)
0002      YY=XX+1.0/XX
0003      WRITE(3, LIST) Z, XX, YY
0004      RETURN
0005      END
```

2.000000000	2.000000000	2.500000000 .0		2.000000000	2.500000000
2	2.000000000	2.500000000		2	2.000000000 2.555555555

11. 文関数の実引数と対応する仮引数の個数

(説明) F700では、実引数と仮引数の個数はコンパイル時に検査されています。

```
0001      DIMENSION P(5)
0002      F(P, Q, R)=(P+Q+R)*P*Q+R
0003      FF=F(1.0, 2.0)
0004      WRITE(3, LIST) FF
0005      STOP
0006      END
```

.2904694011+255

```
ISN  DIAGNOSTIC MESSAGE
0003 268 WRONG NUMBER OF
      ARGUMENTS FOR
      STATEMENT FUNCTION
      "F"
      NUMBER OF FATAL
      ERRORS: 00001
LFSOII MAINPG HAS FATAL ERRORS
```

12. 文関数の実引数と対応する仮引数の型

(説明) F700においては、対応する引数の型が一致しない場合は、エラー・メッセージが出力されます。FL(500)においては、自動的に変換されているようです。

```

0001      F(P,N)=(P+2.0)**N
0002      FF=F(1.0,3.0)
0003      WRITE(3,LIST) FF
0004      STOP
0005      END

```

27.00000000

```

ISN  DIAGEOSTIC MESSAGE NO
0002 272 WRONG TYPE OF ARG-
      ARGUMENT FOR STATE-
      MENT FUNCTION "F"
      NUMBER OF FATAL
      ERRORS: 00001
LFSOII MAINPG HAS FATAL ERRORS

```

13. 文関数 $F(x, y) = e$ において、 e の要素として配列要素が入ってもよいか

(説明) FL(500) のマニュアルによりますと、許されないことになっていて、エラー・メッセージが出力されますが、実際には、許されています。

```

0001      DIMENSION A(2)
0002      DATA A(1)/1.0/, A(2)/2.0/
0003      F(P,Q)=A(1)*P+A(2)*Q+P*Q
0004      FF=F(3.0,4.0)
0005      WRITE(3,LIST) A, FF
0006      STOP
0007      END

```

```

SEQ      DIAGNOSTIC MESSAGE      1.000000000  2.000000000  23.000000000
0003 122 WARNING: ARRAY ELEMENT OF
      "A" USED IN STATEMENT FUN-
      CTION EXPRESSION...ACCEPTED
0003 122 WARNING: ARRAY ELEMENT OF
      "A" USED IN STATEMENT FUNC-
      TION EXPRESSION...ACCEPTED
1.000000000  2.000000000  23.000000000

```

14. 文関数定義文と実行文の順序

(説明) FL(500) では、実行文のあとでも、許されていますが、F700 ではエラー・メッセージが出力されます。(CALL RENAME 文は、実行文です。)

```

0001      DIMENSION A(5)
0002      CALL RENAME (5,:SIU:)
0003      CALL RENAME(6,:SPR:)
0004      F(X,Y)=(X+Y)*(X+Y)+1.0
0005      XF=3.0
0006      YF=4.0
0007      A(1)=F(XF,YF)
0008      WRITE(6,LIST) A, XF, YF
0009      STOP
0010      END

```



```

50.00000000 .0000000000 .0000000000
.0000000000 .0000000000 3.0000000000
4.0000000000

```

```

ISN DIAGNOSTIC MESSAGE
0004 240 ILLEGAL NAME USED
        ON LEFT SIDE OF
        ASSIGNMENT STATEMENT
0007 256 VARIABLE NAME "F"
        FOLLOWED BY LEFT
        PARENTHESES
0007 286 ILLEGAL EXPRESSION
        SYNTAX
        NUMBER OF FATAL ERR
        ERRORS: 00003

025 WARNING: UNDEFINED
    VARIABLE "X"
025 WARNING: UNDEFINED
    VARIABLE "Y"
    NUMBER OF WARNINGS:
    00002
LFS01I MAINPG HAS FATAL ERRORS

```

15. 名前つき COMMON に初期値を設定するとき

(説明) p. 42プログラミング上の注意 1 をご覧下さい。

```

0001      COMPLEX C
0002      DOUBLE PRECISION D
0003      COMMON /BK/A, B, C/BD/D
0004      DIMENSION A(2)
0005      DATA A, B, C/3 * 1.0, (1.0, -1.0)/
0006      DATA D/1.0D0/
0007      ABD=A(1)+B+D
0008      AB=A(2)+B+D
0009      WRITE(3,LIST) ABD, AB, C
0010      STOP
0011      END

```

```

3.000000000 3.000000000 1.000000000
-1.000000000

```

```

ISN DIAGNOSTIC MESSAGE
537 LABELED COMMON BLOCK
    INITIALIZED IN NON-BLOCK
    DATA SUBPROGRAM
537 LABELED COMMON BLOCK
    INITIALIZED IN NON-BLOCK
    DATA SUBPROGRAM
537 LABELED COMMON BLOCK
    INITIALIZED IN NON-BLOCK
    DATA SUBPROGRAM
537 LABELED COMMON BLOCK IN
    INITIALIZED IN NON-BLOCK
    DATA SUBPROGRAM
    NUMBER OF FATAL ERRORS:
    00004
LFS01I MAINPG HAS FATAL ERRORS
LFS10I MAINPG      IS NOT PUT ON MGO

```

16. 8 進型データの中に空白が入ってもよいか

(説明) FL(500)では、許されていませんが、F700では、見易いように空白を入れる
空白を入れることができます。

```
A=11012345670123
B=1101234567 0123
WRITE(3, LIST) A, B
STOP
END
```

```
SEQ      DIAGNOSTIC MESSAGE | .3265295222  .3265295222
0002 006 ILLEGAL CHARACTER IN
        OCTAL STRING
0002 090 SYNTAX ERROR IN CON-
        STANT
0002 108 ILLEGAL EXPRESSION
        SYNTAX
FATAL ERRORS PREVENT SUCCESSFUL
LINKLOADING AND EXECUTION
```

17. 基本外部関数の種類

基本外部関数	英 字 名	FORTRAN L	FORTRAN 700
指 数	EXP	○	○
	DEXP	○	○
	CEXP	○	○
	DCEXP (CDEXP)	×	○
	EXP2	×	○
	DEXP2	×	○
自 然 対 数	EXP10	×	○
	DEXP 10	×	○
	ALOG	○	○
	DLOG	○	○
常 用 対 数	CLOG	○	○
	DCLOG (CDLOG)	×	○
	ALOG10	○	○
	DLOG10	○	○
正 弦	SIN	○	○
	DSIN	○	○
	CSIN	○	○
	DCSIN (CDSIN)	×	○
	COS	○	○
余 弦	DCOS	○	○
	CCOS	○	○
	DCCOS (CDCOS)	×	○
	TAN	×	○
正 接	DATAN	×	○
	COTAN (COT)	×	○
余 接	DCOTAN (DCOT)	×	○
	ARSIN	×	○
逆 正 弦	DARSIN	×	○
	ARCOS	×	○
逆 余 弦	DARCOS	×	○
	ATAN	○	○
逆 正 接	DATAN	○	○
	ATAN2	○	○
	DATAN2	○	○
	DATAN2	○	○

基本外部関数	英字名	FORTRAN L	FORTRAN 700
双曲線正弦	SINH	×	○
	DSINH	×	○
双曲線余弦	COSH	×	○
	DCOSH	×	○
双曲線正接	TANH	○	○
	DTANH	×	○
正弦 (単位度)	SIND	×	○
余弦 (単位度)	COSD	×	○
平方根	SQRT	○	○
	DSQRT	○	○
	CSQRT	○	○
	DCSQRT	×	○
	(CDSQRT)		
立法根	CBRT	×	○
	DCBRT	×	○
剰余	DMOD	○	○
絶対値	CABS	○	○
	DCABS (CDABS)	×	○
偏角	CANG	×	○
最大公約数	IGCM	×	○
指数部取出し	IRE	×	○
	IDE	×	○
数値部取出し	AMT	×	○
	DMT	×	○
誤差関数	ERF	×	○
	DERF	×	○
補誤差関数	ERFC	×	○
	DERFC	×	○
ガンマ関数	GAMMA	×	○
	DGAMMA	×	○
対数ガンマ関数	ALGAMA	×	○
	DLGAMA	×	○
一様乱数	RANDOM	×	○

基本外部関数	英 字 名	FORTRAN L	FORTRAN 700
ビット関数	ISHIFT	×	○
	LSHIFT	×	○
	ITRFOM	×	○
	ISCAN	×	○
	IBTCNT	×	○
	ISETBT	×	○
	ICVTBT	×	○

注) 英字名の欄の中のかっこでかこまれた英字名は、その基本外部関数をかっこでかこまれた英字名によっても引用できることを意味する。

組込み関数

組込み関数の種類

組込み関数	英 字 名	FORTRAN L	FORTRAN 700
絶 対 値	ABS	○	○
	ABS	○	○
	DABS	○	○
切 捨 て	AINT	○	○
	INT	○	○
	IDINT	○	○
剰 余	AMOD	○	○
	MOD	○	○
最 大 値	AMAX0	○	○
	AMAX1	○	○
	MAX0	○	○
	MAX1	○	○
	DMAX1	○	○
最 小 値	AMIN0	○	○
	AMIN1	○	○
	MIN0	○	○
	MIN1	○	○
	DMIN1	○	○

組込み関数	英 字 名	FORTRAN L	FORTRAN 700
実 数 化	FLOAT	○	○
倍精度実数化	DFLOAT	○	○
整 数 化	IFIX	○	○
符号のつけかえ	SIGN	○	○
	ISIGN	○	○
	DSIGN	○	○
起 過 分	DIM	○	○
	IDIM	○	○
	DDIM	×	○
単 精 度 化	SNGL	○	○
実 数 部	REAL	○	○
	DREAL	×	○
虚 数 部	AIMAG	○	○
	DIMAG	×	○
倍 精 度 化	DBLE	○	○
複 素 数 化	CMPLX	○	○
	DCMPLX	×	○
共役複素数化	CONJG	○	○
	DCONJG	×	○
論 理 積	IAND	○	○
論 理 和	IOR	○	○
論 理 否 定	ICOMPL	○	○
排他的論理和	IEXCLR	○	○
論 理 演 算	IBOOL	×	○

プログラミング上の注意

1. BLOCK DATA の使用について

名前つき COMMON に初期値を設定する場合、FORTRAN L-500 では、DATA 文でできましたが、FORTRAN 700 の場合、BLOCKDATA 文を使用して下さい。

(例)

```
0001 COMPLEX
0002 DOUBLE PRECISION D
0003 COMMON /BK/A, B, C/BD/D
0004 DIMENSION A(2)
0005 ABD=A(1)+ B+D
0006 AB=A(2)+B+D
0007 WRITE (3, LIST) ABD, AB, C
0008 STOP
0009 END
```

① TITLEBLOCKD (第2カラムから)

```
0001 BLOCK DATA
0002 COMMON /BK/A, B, C/BD/B
0003 DIMENSION A(2)
0004 COMPLEX C
0005 DOUBLE PRECISION D
0006 DATA A, B, C/3*1.0, (1.0, -1.0)/
0007 DATA D/1.0D0/
0008 END
```

② ENDS△B

①は、これ以降のカードが、初期値設定副プログラムであることを宣言するために必要です。

②は、このカードにより、システムの方でBLOCK DATAを扱うための必要なコントロールカードが展開されるようになっています。

2. FORTRAN 700 で、OPTION OVF を指定したときの 0 除算

ISN	LABEL FORTRAN STATEMENT
0001	A=2.0
0002	B=0.0
0003	C=A/B
0004	WRITE(3, LIST) A, B, C
0005	STOP
0006	END

```

**401N ZERO DIVIDE
      ERROR DETECTED IN NFSANCHOR          00010064
      CALLED FROM MAINPG          0003      00003450
2.000000000          .0          .0

```

3. FORTRAN 700 では、DIMENSION を宣言をしますと、システム側で0 クリアされています。

```

      C      DIMENSION CLEAR TEST (72/10/27)
0001      DIMENSION AS(10)
0002      WRITE(3, LIST) AS
0003      STOP
0004      END

.0      .0      .0      .0      .0      .0      .0
.0      .0      .0

```

4. 引数が配列名であるときの DIMENSION 宣言の内部的な相違について

```

メイン DIMENSION P(3), Q(4), R(5)
      {
      CALL POPY(P, Q, R)
      }
      STOP
      END

サブ SUBROUTINE POPY(PD, QD, RD)
      DIMENSION PD(1), QD(1), RD(1)
      DO 10 I=1, 3
      (10 PD(I)=FLOAT(I)
      {
      RETURN
      END

```

引数の受け渡しは、引数の格納されている番地（アドレス）によるので、メインにおいて宣言された配列の大きさを越えることがなければ正常に実行されます。

5. SSL の SRNDOM1 によって発生する乱数は [0, 1] の値です。

6. DO の拡張範囲の具体例

次の DO の拡張は、正常に実行されます。


```

0001      DIMENSION LL(10)
0002      DO 5 I=1,10
0003      5  LL(I)=1
0004      K=0
0005      DO 1 I=1,100
0006      WRITE(3,100) I
0007      100 FORMAT(2H*,I6)
0008      J=I-I/3*3
0009      IF(J.EQ.0) GO TO 2
0010      1  CONTINUE
0011      K=1
0012      2  WRITE (3,200) I, (LL(L), L=1,10)
0013      200 FORMAT(3H* *, I4,3H *,10I5)
0014      IF(K,EQ.0) GO TO 1
0015      STOP
0016      END

```

7. 入力のプログラム・ミスについて

```

(例)      }
          READ(2,11) A
          11 F0RMA(T(F10.1)
          }
          END

```

データ

0.123 E8

↑

1 カラム

全体が10桁になるように0が3つ補われて、0.123E8000 と読みこまれて、オーバー・フローします。 0.123E8000

8. F0RTRAN 700 では、添字の値は次のようになっています。

添字の値

次元の数	宣言子添字	添字	添字の値	添字の最大値
1	(A)	(a)	a	A
2	(A,B)	(a,b)	a+A(b-1)	A·B
3	(A,B,C)	(a,b,c)	a+A·(b-1)+A·B·(c-1)	A·B·C
4	(A,B,C,D)	(a,b,c,d)	a+A·(b-1)+A·B·(c-1)+A·B·C·(d-1)	A·B·C·D
5	(A,B,C,D,E)	(a,b,c,d,e)	a+A·(b-1)+A·B·(c-1)+A·B·C·(d-1) +A·B·C·D·(e-1)	A·B·C·D·E
6	(A,B,C,D,E,F)	(a,b,c,d,e,f)	a+A·(b-1)+A·B·(c-1)+A·B·C·(d-1) +A·B·C·D·(e-1)+A·B·C·D·E·(f-1)	A·B·C·D·E·F
7	(A,B,C,D,E,F,G)	(a,b,c,d,e,f,g)	a+A·(b-1)+A·B·(c-1)+A·B·C·(d-1) +A·B·C·D·(e-1)+A·B·C·D·E·(f-1) +A·B·C·D·E·F·(g-1)	A·B·C·D·E·F·G

ここで、a,b,c,d,e,f,gは添字式とし、A,B,C,D,E,F,Gは寸法を示す。

文法に関する補足とその他の注意

FÖRTRAN700 プログラミング説明書からの転載です。次回に説明を補いたいと思います。

以下に文法説明書に対する補足事項を述べる。各項目の前にある数実は、文法説明書の章節番号である。

- (1) END行に文の番号があるとコンパイル時に致命的なエラーとなる。…………… (文法3.2.1)
- (2) プログラム単位の最初の行が継続行であれば、その行は開始行とみなされる。
…………… (文法3.2.4)
- (3) 注釈行の直後の継続行は、注釈行の前にある開始行あるいは継続行の継続行とみなされる。
…………… (文法3.2.4)
- (4) 1つの文の中に含まれる文字数は、文字定数中の空白を除き、空白をとり除いた後の文字数が1320字まで許される。…………… (文法3.2.4)
- (5) 1つのプログラム単位中に書ける文の数は最大4095である。END行も1つの文として数えられる。この制限を越えるとコンパイル時に致命的なエラーとなる。…………… (文法3.3)
- (6) 割当て形 GO TO 文において、整数型変数名にそのGO TO 文中の文の番号の並びにない文の番号が割当てられても、その割当てられた文の番号をもつ文に実行の制御が渡る。エラー・メッセージは出されない。…………… (文法7.1.2(1)(b))
- (7) 計算形 GO TO 文において、整数型変数に範囲外の値が入れられていると、実行時にエラー・メッセージが出されUEPになる。…………… (文法7.1.2(1)(c))
- (8) CALL 文の実引数は、文の番号を除き最大63個まで許される。この制限を越えるとコンパイル時に致命的エラーとなる。実引数として使われる文の番号の個数は63以下でなければならない。…………… (文法7.1.2(4))
- (9) RETURN_i 文の *i* の値が許される範囲を越えていると実行時にエラー・メッセージが出されUEPになる。許される範囲はつぎの通りである。
実引数となっている文の番号の個数： ℓ
仮引数となっている星印(*)の個数： m
 $1 \sim \min(\ell, m)$ が許される範囲である。 ℓ キ m であっても、エラー・メッセージは出力されない。
…………… (文法7.1.2(5))
- (10) PAUSE 文あるいはSTOP 文の実行によって、コンソール・タイプライタ上にメッセージが出力される。…………… (文法1.1.2(5))
- (11) DO 文は最大63個の入れ子をなしてもよい。この制限を越えるとコンパイル時に致命的エラーとなる。(7.1.3(2)(2)) を参照されたい。
- (12) DO の拡張範囲に DO 文があつたり、DO の範囲外からDO の範囲内に実行の制御が渡ってもエラーは検出されない。実行時に不確定な動作を起す。…………… (文法7.1.2(8))

- (13) 入出力文におけるDO形並びは最大63重まで許される。DO文の範囲内にある入出力文における場合は、DOの入れ子の回数と合計したものが63重を越えてはならない。
 (文法7.1.3(2)(2))
- (14) 1つのプログラム単位内に同一ファイルに対して2回以上DEFINE FILE文による定義があってもコンパイル時にはエラーとはみなされない。実行時に1つの実行プログラム内で同一ファイルに対して2回以上DEFINE FILE文が実行されると、2回目以降の実行は無視される。..... (文法7.1.5(1))
- (15) 定数の添字式のみをもつ配列要素の添字の値は、その定数宣言子添字の値を越えていれば、配列名が仮引数で1次元1と宣言されている場合をのぞき、コンパイル時に致命的エラーとなる。その他の場合の配列要素の添字の値の範囲に関するエラーは実行時にSUBCHK文によってのみ調べられる。..... (文法7.2.2)
- (16) 同じ名前をもつ名前付き共通ブロックの大きさが異なると(別の2つのプログラム単位中で定義されたもの)、リンクロード時にエラー・メッセージが出される(EXTENDオプションを指定したのみ)。ただし先にリンクロードされたプログラム単位中で定義された名前付き共通ブロックの方が大きければエラー・メッセージは出されず、小さい方は大きい方の記憶場所の一部を共有する。..... (文法7.3(2))
- (17) DATA文のDO形並びは、最大7重の入れ子をなしていてもよい。..... (文法7.4)
- (18) 欄記述子における欄の幅等について、つぎの様に調べられる。
 $Iw, Fw, d, Ew, d, Gw, d, Lw, Aw, nHh_1h_2 \cdots h_n$ (または: $h_1h_2 \cdots h_n$: あるいは $h_1h_2 \cdots h_n$), Ow, Bw, nX, Tp に対して w, d, p, n のいずれかが133を越えていればコンパイル後に警告メッセージが出力される。
 つぎの場合は致命的なエラーとみなされる。
 w, p, n あるいは r が4095を越えるか、ゼロ。 d が4095を越える。
 桁移動子が-2048~2047の範囲外。..... (文法7.5.1)
- (19) 実行時においては書式により定められた記録の長さが、ファイル・テーブル・サブプログラムにより定められた入出力バッファの長さを越えていれば、エラー・メッセージが出力されUEPになる。..... (文法7.5.1)
- (20) 手続きの仮引数の個数は最大63個まで許される。..... (文法8)
- (21) 引用側の実引数の個数と仮引数の個数は一致しなければならない。外部手続きの仮引数の個数と対応する実引数の個数が一致しなければリンクロード時に致命的エラーとなる(エラー・メッセージ LLK 66I) (文法8)
- (22) 外部手続きの型と引用側の型とは一致しなければならない。型が一致しなければリンクロード時に致命的エラーとなる。(エラー・メッセージ LLK 66I) (文法8)
- (23) 実引数の型と対応する仮引数の型との一致は検査されない。実行時には、実引数の型の内部表現が仮引数の型の内部表現とみなされて実行される。..... (文法8)

- (24) 基本外部サブルーチンの引用における実引数の型が、決められている型と一致しなければ、実引数の型の変換は行なわない。…………… (文法 8.6)
- (25) SUBTRACE 文に書かれた名前は外部手続き名かどうか調べられない。…… (文法 11(8))
- (26) 2つの DOCHK 文により指定された範囲が重複してはならない。……………(文法 11(9))
- (27) 1つのプログラム内で配列要素の参照の回数 DO 文の算術演算子を含む初期値パラメータ終値パラメータおよび増分パラメータの個数の合計が4095を越えると、コンパイルは中断される。…………… (その他)

FORTRAN 700 の手続き名とサブプログラム名の対応

手 続 き 名	サブプログラム名	手 続 き 名	サブプログラム名
基本外部サブルーチン		AMINI	AMINI
CHAIN	NFSCCHAIN	AMOD	AMOD
CLOCK	CLOCK	AMT	AMT
CPTIME	CPTIME	ARCOS ⁽³⁾	ARCOS
DATE	DATE	ARSIN	NFSDARSIN
DECODE	DECODE	ATAN2 ⁽³⁾	ATAN2
DUMP	DUMP	ATAN	ATAN
DVCHK	DVCHK	CABS ⁽³⁾	CABS
ERRCNT	ERRCNT	CANG	CANG
ENCODE	ENCODE	CBRT	CBRT
EXIT	EXIT	CCOS ⁽³⁾	CCOS
MDUMP	MDUMP	CEXP ⁽³⁾ CEXP ⁽³⁾	CEXP
OVERFL	OVERFL	CLOG ⁽³⁾	CLOG
PDUMP	PDUMP	COS ⁽³⁾	COS
PHASEL	PHASEL	COSD ⁽³⁾	COSD
PHASER	PHASER	COSH ⁽³⁾	COSH
RENAME	RENAME	COTAN	COTAN
REREAD	REREAD	(COT)	
SLITE	SLITE	複素数除算	CDIV
SLITET	SLITET	複素数乗算	CMUL
SSWTCH	SSWTCH	CSIN ⁽³⁾	CSIN
TIME	TIME	CSQRT ⁽³⁾	CSQRT
UNLABL	UNLABL	DARCOS ⁽³⁾	DARCOS
基本外部関数・組込み関数		DARSIN	DARSIN
AINT	NFSDAINT	DATAN2 ⁽³⁾	DATAN2
ALGAMA ⁽³⁾	ALGAMA	TIME	DATAN
ALOG10 ⁽³⁾	ALOG10	DCABS ⁽³⁾	DCABS
ALOG	ALOG	(CDABS)	
AMAX0	AMAX0	DCBRT	DCBRT
AMAX1	AMAX1	DCCOS ⁽³⁾	DCCOS
AMIN0	AMIN0	(CDCOS)	

手 続 き 名	サブプログラム名	手 続 き 名	サブプログラム名
倍精度複素数除算	DCDIV	EXPDR ⁽¹⁾⁽³⁾	EXPDR
DCEXP ⁽³⁾	DCEXP	EXPID ⁽¹⁾⁽³⁾	EXPID
(CDEXP)		EXPII ⁽¹⁾	NFSDEXPII
DCLOG ⁽³⁾	DCLOG	EXPIR ⁽¹⁾⁽³⁾	EXPIR
(CDLOG)		EXPRD ⁽¹⁾⁽³⁾	EXPRD
倍精度複素数乗算	DCMUL	EXPRI ⁽¹⁾⁽³⁾	EXPRI
DCOS ⁽³⁾	DCOS	EXPRR ⁽¹⁾⁽³⁾	EXPRR
DCOSH ⁽³⁾	DCOSH	GAMMA	GAMMA
DCOTAN	NFSDDCOTAN	IBTCNT	IBTCNT
(DCOT)		ICVTBT	ICVTBT
DCSIN ⁽³⁾	DCSIN	IDE	IDE
(CDSIN)		IDIM	IDIM
DCSQRT ⁽³⁾	DCSQRT	IDINT	IDINT
(CDSQRT)		IFIX	IFIX
DDIM	DDIM	IGCM	IGCM
DERF	DERF	INT	INT
DERFC ⁽³⁾	DERFC	IRE	IRE
DEXP10 ⁽³⁾	DEXP10	ISCAN	ISCAN
DEXP2 ⁽³⁾	DEXP2	DLOG	ISETBT
DEXP	DEXP	ISHIFT	ISHIFT
DGAMMA	DGAMMA	ISIGN	ISIGN
DIM	DIM	ITRFOM	ITRFOM
DLGAMA ⁽³⁾	DLGAMA	LSHIFT	LSHIFT
DLOG10	DLOG10	MAX0	MAX0
DLOG	DLOG	MAXI ⁽³⁾ 1	MAX1
DMAX1	DMAX1	MIN0	MIN0
DMIN1	DMIN1	MINI ⁽³⁾	MIN1
DMOD	DMOD	MOD	MOD
DMT	DMT	RANDOM ⁽²⁾	RANDOM
DSIGN	DSIGN	SIGN	SIGN
DSIN	DSIN	SIN	SIN
DSINH ⁽³⁾	DSINH	SIND ⁽³⁾	SIND
DSQRT	DSQRT	SINH ⁽³⁾	SINH
DTAN	DTAN	SQRT	SQRT
DTANH ⁽³⁾	DTANH	TAN	TAN
ERF	ERF	TANH ⁽³⁾	TANH
ERFC ⁽³⁾	ERFC	external 用	
EXP10 ⁽³⁾	EXP10	ABS	NFSEABS
EXP2 ⁽³⁾	EXP2	AIMAG	AIMAG
EXP	EXP	AINT	AINT
EXPCI ⁽¹⁾	EXPCI	ALGAMA ⁽³⁾	NFSEALGAMA
EXPDCI ⁽¹⁾	EXPDCI	ALOG10 ⁽³⁾	ALOG10
EXPDD ⁽¹⁾⁽³⁾	EXPDD	ALOG	ALOG
EXPDI ⁽¹⁾⁽³⁾	EXPDI	AMOD	AMOD

手 続 き 名	サブプログラム名	手 続 き 名	サブプログラム名
AMT	AMT	(CDSQRT)	
ARCOS ⁽³⁾	ARCOS	DDIM	DDIM
ARSIN	ARSIN	DERF	DERF
ATAN2 ⁽³⁾	ATAN2	DERFC ⁽³⁾	DERFC
ATAN	ATAN	DEXP10 ⁽³⁾	DEXP10
CABS ⁽³⁾	CABS	DEXP2 ⁽³⁾	DEXP2
CANG	CANG	DEXP	DEXP
CBRT	CBRT	DFLOAT	DFLOAT
CCOS ⁽³⁾	CCOS	DGAMMA	DGAMMA
CEXP ⁽³⁾	CEXP	DIM	DIM
CLOG ⁽³⁾	CLOG	DIMAG	DIMAG
CMPLX	CMPLX	DLGAMA ⁽³⁾	DLGAMA
CONJG	CONJG	DLOG10 ⁽³⁾	DLOG10
COS ⁽³⁾	COS	DLOG	DLOG
COSD ⁽³⁾	COSD	DMOD	DMOD
COSH ⁽³⁾	COSH	DMT	DMT
COTAN	COTAN	DREAL	DREAL
(COT)		DSIGN	DSIGN
CSIN ⁽³⁾	CSIN	DSIN	DSIN
CSQRT ⁽³⁾	CSQRT	DSINH ⁽³⁾	DSINH
DABS	DABS	DSQRT	DSQRT
DARCOS ⁽³⁾	DARCOS	DTAN	DTAN
DARSIN	DARSIN	DTANH ⁽³⁾	DTANH
DATAN2 ⁽³⁾	DATAN2	ERF	ERF
DATAN	DATAN	ERFC ⁽³⁾	ERFC
DBLE	DBLE	EXP10 ⁽³⁾	EXP10
DCABS ⁽³⁾	DCABS	EXP2 ⁽³⁾	EXP2
(CDABS)		EXP	NFSEEXP
DCBRT	DCBRT	FLOAT	FLOAT
DCCOS ⁽³⁾	DCCOS	GAMMA	GAMMA
(CDCOS)		IABS	IABS
DCEXP ⁽³⁾	DCEXP	IAND	IAND
(CDEXP)		IBOOL	IBOOL
DCLOG ⁽³⁾	NFSEDCLOG	IBTCNT	IBTCNT
(CDLOG)		ICOMPL	ICOMPL
DCMPLX	DCMPLX	ICVTBT	ICVTBT
DCONJG	DCONJG	IDE	IDE
DCOS ⁽³⁾	DCOS	IDIM	IDIM
DCOSH ⁽³⁾	DCOSH	IDINT	IDINT
DCOTAN	DCOTAN	IEXCLR	IEXCLR
(DCOT)		IFIX	IFIX
DCSIN ⁽³⁾	DCSIN	IGCM	IGCM
(CDSIN)		INT	INT
DCSQRT ⁽³⁾	DCSQRT	IOR	IOR

手 続 き 名	サブプログラム名	手 続 き 名	サブプログラム名
IRE	IRE	REAL	REAL
ISCAN	NFSEISCAN	SIGN	SIGN
ISETBT	ISETBT	SIN	SIN
ISHIFT	ISHIFT	SIND ⁽³⁾	SIND
ISIGN	ISIGN	SINH ⁽³⁾	SINH
ITRFOM	ITRFOM	SNGL	SNGL
LSHIFT	LSHIFT	SQRT	SQRT
MOD	MOD	TAN	TAN
RANDOM ⁽³⁾	RANDOM	TANH ⁽³⁾	TANH

- 注：(1) これらは外部手続き名ではなく、巾乗演算の組合せを表わすものである。巾乗演算は内部的には外部手続きによって処理される。
- (2) 基本外部関数RANDOMは1度引用されると、その最後の引用が終了する迄、再ロードによって初期状態に戻されるようなフェーズ構成にはならない。
- (3) これらの手続きは、他の手続きを内部的に引用している。引用されている手続きは、それを引用している手続きと同一チェーン内になければならない。引用されている手続きはつぎの通りである。

他の手続きを引用する手続

手 続 き 名	引用される手続き名	手 続 き 名	引用される手続き名
DUMP	PDUMP	DCEXP	DSIN, DCOS, DEXP
MDUMP	PDUMP	(DCEXP)	
ALGAMA	GAMMA	DCLOG	DLOG, DATAN2
ALOG10	ALOG	(CDLOG)	
ARCOS	ARSIN	(DCOS)	DSIN
ATAN2	ATAN	DCOSH	DEXP
CABS	SQRT	DCSIN	DSIN, DCOS, DEXP
CCOS	SIN, COS, EXP	DCSQRT	DSQRT, DCABS
CEXP	SIN, COS, EXP	(CDSQRT)	
CLOG	ALOG, ATAN2	DERFC	DERF
COS	SIN	DEXP10	DEXP
COSD	COS	DEXP2	DEXP
COSH	EXP	DLGAMA	DGAMMA
COTAN	TAN	DLOG10	DLOG
(COT)		DSINH	DEXP
CSIN	SIN, COS, EXP	DTANH	DEXP
CSQRT	SQRT, CABS	ERFC	ERF
DARCOS	DARSIN	EXP10	EXP
DATAN2	DATAN	EXP2	EXP
DCABS	DSRQT	EXPDD	DEXP, DLOG
(CDABS)		EXPDI	EXPDD
DCCOS	DSIN, DCOS, DEXP	EXPDR	EXPDD
(CDCOS)		EXPID	EXPDD

手続き名	引用される手続き名	手続き名	引用される手続き名
EXPIR	EXPRR	MIN1	IFIX
EXPRD	EXPDD	SIND	SIN
EXPRI	EXPRR	SINH H	EXP
EXPRR	EXP, ALOG	TANH	EXP
MAX1	IFIX		