情報処理　（第12回）　(2017/7/3・茂木)

本日のテーマ

- 書式指定・リダイレクトによる入出力
- 配列の演習

演習の解答例

Q.1 内積の計算 (array1.f)

```
c23456
      program array1
      implicit none
      real s, a(3), b(3)
      integer j
      data a/1.0,2.0,3.0/, b/3*1.0/

      s=0.0
      do j=1, 3
         s = s+a(j)*b(j)
      enddo

      write(6,*)'a と b の内積は',s,'です. '
      end
```

Q.2 要素の和 (array2.f)

```
c23456
      program array2
      implicit none
      integer i, j
      real D, a(3,3), s
      data a/1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0/

      s=0.0
      do i=1, 3
         do j=1, 3
            s=s+a(i,j)
         enddo
      enddo

      write(6,*)'a(1,1) から a(3,3) までの和は',s,'です. '
      end
```

Q.3 行列の和の計算 (Q3.f)

```
 1:c23456
 2:       program Q3
 3:       implicit none
 4:       integer j, k
 5:       real a(3,3), b(3,3), C(3,3)
 6:       data a/1.0, 4.0, 7.0, 2.0, 5.0, 8.0, 3.0, 6.0, 9.0/
 7:       data b/11.0, 14.0, 17.0, 12.0, 15.0, 18.0, 13.0, 16.0, 19.0/
 8:
 9:       do j=1, 3
10:          do k=1, 3
11:             C(j,k)=A(j,k)+B(j,k)
12:          enddo
13:       enddo
14:
15:       do j=1, 3
16:           write(6,7000)j, (C(j,k),k=1,3) ! 出力
17:       enddo
18: 7000 format(i2,'行目', f10.3, f9.2, f8.1)
19:       end
```

Q.4 行列とベクトルのかけ算 (array3.f)

```
 1:c23456
 2:       program array3
 3:       implicit none
 4:       integer i, j
 5:       real A(3,3), x(3), y(3)
 6:
 7:       write(6,*)'3x3 行列 A とベクトル x を入力して下さい'
 8:       read(5,*) ((A(i,j),j=1,3), i=1,3)    ! DO 並びで行列とベクトルを読込
 9:       read(5,*) (x(i),i=1,3)               !  i,j の順番にも注意
10:
11:       do i=1, 3
12:          y(i)=0.0
13:          do j=1, 3
14:            y(i)=y(i)+A(i,j)*x(j)
15:          enddo
16:       enddo
17:
18:       write(6,5000) (y(i), i=1, 3)  ! DO 並びで結果を出力
19: 5000 format(f10.2)
20:       end
```

1　リダイレクトによるデータの入力

プログラム array3.f を実行すると，2 つの read 文に対してそれぞれデータを入力する必要があります．一つ一つ手で入力しても構いませんが，データが大きくなると入力も面倒で，かつ，入力ミスも生じます．

そこで，入力するデータをあらかじめ別のファイルに入力しておいて，プログラムの実行時にそのファイルからデータを読み込むようにすると便利です．これは，コマンドプロンプトの「リダイレクト」という機能を使うと簡単で便利です（より高機能のファイル入出力は後日説明します）．

下のように，**入力データファイルをメモ帳で打ち込みます**．このファイルは Fortran のプログラムではないので，左に空白を入れる必要もありませんし，ファイル名（この例では in.txt）も自由です．

入力データファイル in.txt

```
1.0, 2.0, 3.0          ←行列 A
4.0, 5.0, 6.0
7.0, 8.0, 9.0
1.0, 2.0, 3.0          ←ベクトル x
```

array3.f の実行－リダイレクトの使用

```
C:¥work¥T66160¥20170703>f77 array3.f -o array3     ←コンパイル（普段通り）

C:¥work¥T66160¥20170703>array3 < in.txt  ←実行・データは in.txt から読み込む.
 3x3 行列 A とベクトル x を入力して下さい ←write(6,*) による出力は普段通り画面へ
     14.00
     32.00
     50.00
```

2　書式指定 (format specification・教科書 p.95-102)

今までのように read(5,*)，write(6,*) として入出力すると，出力される欄の幅がそろわず結果が見にくくなることがあります．このため，format 文を使って入出力されるデータの表現形式を指定（**書式指定**）することができます．

解答例の Q3.f のプログラム中 16 行目

```
        write(6,7000)j, (C(j,k),k=1,3) ! 出力
```

の 7000 は書式指定を記した format 文の文番号を表し，

```
 7000 format(i2,'行目', f10.3, f9.2, f8.1)
```

の書式で j と an(j,1)，an(j,2)，an(j,3) を出力することを指定しています．具体的には，まず，j の値を I2 の書式（整数，2 桁）で出力し，次に,「行目」という文字を出力した後，an(j,1) を F10.3(実数，小数点以下 3 桁，全体 10 文字分），an(j,2) を F9.2(実数，小数点以下 2 桁，全体 9 文字分），an(j,3) を F8.1(実数，小数点以下 1 桁，全体 8 文字分）で出力します．従って次のような出力が得られます．

```
 1行目     12.000     14.00     16.0
 2行目     18.000     20.00     22.0
 3行目     24.000     26.00     28.0
12     12345678901234567891234567 8 (この行は説明用で実際には出力されません)
```

なお，以下のように do 並びを用いて，かつ，書式を付けて横に並べて出力するデータ数をコントロールすることで，プログラムの出力部分をコンパクトに書くことができます（15〜17 行目の 3 行を次の 1 行に置き換える）．

```
15:       write(6,7000)(j,(C(j,k),k=1,3), j=1,3)
```

I2，F10.3 などは編集記述子 (edit descriptor) と呼び，様々な種類があります（サンプルプログラムでは I，F が大文字でなく小文字になっていますが，どちらでもかまいません．以降では慣例に従って大文字で書きます）．

2.1 編集記述子

編集記述子 (edit descriptor) は，その記号の前に，反復回数（何回その書式で繰り返し入出力するか）を付けることのできる (1) 反復可能編集記述子 (repeatable edit descriptor) と，それが指定できない (2) 反復不可能編集記述子 (nonrepeatable edit descriptor) に分けられます.

反復可能編集記述子

整数の入出力	Iw	I$w.m$
実数の入出力	F$w.d$	
同上	E$w.d$	E$w.d$Ee
同上	D$w.d$	
同上	G$w.d$	G$w.d$Ee
論理型の入出力	Lw	
文字型の入出力	A	Aw

I, F, E, D, G, L, A は編集方法，w，e は 1 以上の整定数，d，m は 0 以上の整定数

反復不可能編集記述子

文字（列）の出力	'$h_1h_2\cdots h_n$'	nH$h_1h_2\cdots h_n$	
位置づけ編集	Tc	TLc	TRc
位置づけ編集	nX		
改行	/		
文字の入出力制御	:		
正符号の制御	S	SP	SS
桁移動	kP		
数値入力時の空白の解釈	BN	BZ	

アポストロフィー ('),H,T,X, 斜線 (/), コロン (:), S,SP, SS, P, BN, BZ は編集方法，h は 1 文字，n,c は 1 以上の整定数，k は正負の整定数

2.2 整数値の書式指定 Iw, I$w.m$

w は欄の幅（入出力される最大の文字数）であり，w 文字を越えて入力された場合超過した文字が無視されます．また，w 桁で出力できない場合には*で埋められます．I$w.m$ は出力のみに用いられ，最低 m 文字出力されるように 0 を補います.

入力			出力		
編集記述子	キー入力	読み込まれる値	編集記述子	出力する値	得られる表示
I5	␣␣123	123	I5	123	␣␣123
I5	␣␣␣␣␣	0	I5	-123	␣-123
I5	+1␣23	123	I5	-12345	*****
I5	-12345	-1234	I5.3	12	␣␣012

␣ は半角スペースを表す.

2.3 単精度および倍精度実数値の書式指定 F$w.d$, E$w.d$, E$w.d$Ee, D$w.d$

w は欄の幅，d は小数部の桁数，e は指数部の桁数です．入力データに小数部がある場合，入力データの小数点の位置が優先されます．入力データに小数点がない場合，入力文字の右から d 桁が小数部とみなされます．1.0E+01, 1.e1, 1.d01 などのように指数部を付けて入力することもできます.

出力の場合，F$w.d$ 型では，通常の小数の形式で出力されます．このため，符号と小数点のために少なくとも $w \geq d+2$ でなければなりませんが，$w-d$ の値が小さいとすぐに桁あふれを起こして出力に失敗してしまいます．この場合，欄全体に*が出力されます．E$w.d$, D$w.d$ 型では指数を用いて，小数点以下 d 桁に丸めて出力されます．通常，指数の表示が E+00（D$w.d$ 型では D+00）であ

るため，符号と小数点を考えて，$w \geq d+6$ でなければなりません．E$w.d$ の場合にはすべて小数点以下 d 桁に丸めてしまうので，$w \geq d+6$ の条件を満たせば出力に失敗することはありません．

入力			出力		
編集記述子	キー入力	読み込まれる値	編集記述子	出力する値	得られる表示
F7.3	123.456	123.456	F7.3	123	123.000
F7.3	123456	123.456	F7.3	-12.3	-12.300
F7.3	1.0E+01	10.0	F7.3	1234.50	*******
F7.3	1	0.001	F7.0	1.23	␣␣␣␣␣1.
F7.0	-1.2345	-1.2345	E14.6	1.23	␣␣0.123000E+01
F7.0	-1.23456	-1.2345	E10.4	-1.23456	-.1234E+01
E6.2	123456	1234.56	E9.4	-1.23456	*********
E6.2	1.2345	1.23456	E9.4E1	-1.23456	-.1234E+1

Iw 型，F$w.d$ 型の編集記述子の例 sample1.f

```
      program sample1
      real a(2,3)
      read(5,5000) ((a(i,j),j=1,3),i=1,2)      ! ネストされた DO 並びで読込
      write(6,5002) ((i,j,a(i,j),j=1,3),i=1,2)! i,j,a(i,j) を DO 並びで出力
 5000 format(10f7.2)
 5002 format( 3(2i5,':', f7.2) )  ! i5,i5,':',f7.2 の出力を一行に３組出力
      stop
      end
 (実行結果)
C:¥work¥T66160¥20170703>sample1
1.,2.,3.,4.,5.,6.                                    ! キーボードから入力
    1    1:   1.00    1    2:   2.00    1    3:   3.00   ! ２行で出力
    2    1:   4.00    2    2:   5.00    2    3:   6.00
```

2.4 文字型の書式指定 A, Aw

文字型変数 (character variable) の入出力に用います．欄の幅を w で指定します．指定しない場合には文字型変数の文字数になります．以下に文字型変数の使用例を示します．

文字型変数の宣言と入出力 sample2.f

```
c23456
      program lsample2
      character t*20    ! 半角文字 20 文字 (20byte) を記憶できる文字型変数
      read(5,5000) t
 5000 format(a)         ! 欄の幅の指定がないので最大 20 文字読込むことができる
      write(6,5002) t
 5002 format(a10)       ! 先頭 10 文字分出力
      stop
      end
 (実行結果)
C:¥work¥T66160¥20170703>sample2
abcde                   ! 5 文字だけ入力
abcde                   ! 読み込まれた 5 文字が出力される

C:¥work¥T66160¥20170703>sample2
abcdefghijklmnopqrstu   ! 21 文字入力（実際に読み込まれるのは 20 文字）
abcdefghij              ! 先頭の 10 文字だけ出力される
```

2.5 文字列定数の出力　アポストロフィ編集記述子

アポストロフィ（’）で挟んだ文字列をそのまま表示します．nH 型編集記述子でも同様に文字列定数が出力できます．

```
      format('GOKEI=',i5)        ! アポストロフィー編集記述子
      format(6HGOKEI=,i5)        ! nH 型編集記述子
```

2.6 現在の入出力位置を右へ移動　nX 型編集記述子

現在の入出力位置を右へn文字分移動します．結果として，read 文で用いると現在位置からn文字読み飛ばし，write 文で用いるとn個の空白が出力されることになります．

2.7 複数行の書式指定をまとめて記述　斜線編集記述子

斜線編集記述子/によって複数行の書式指定をまとめて指定することができます．1つの書式指定が一行分の入出力に相当するため，これによって複数行にわたる入出力が行えます．

アポストロフィ，斜線，nX 型編集記述子の使用例 sample3.f

```
      program sample3
      integer tanka/200/, kosuu/3/
      write(6,5000) tanka, kosuu, tanka*kosuu
 5000 format('単価=',i10,3x,'個数=',i5/,'合計=',i10)
      stop
      end
（実行結果）
C:¥work¥T66160¥20170703>sample3
単価=       200    個数=     3
合計=       600
```

3 練習問題

1. 九九の計算を行い，以下のような表を kuku.txt に出力するプログラムを作成せよ．

```
 1  2  3  4  5  6  7  8  9
 2  4  6  8 10 12 14 16 18
 3  6  9 12 15 18 21 24 27
 4  8 12 16 20 24 28 32 36
 5 10 15 20 25 30 35 40 45
 6 12 18 24 30 36 42 48 54
 7 14 21 28 35 42 49 56 63
 8 16 24 32 40 48 56 64 72
 9 18 27 36 45 54 63 72 81
```

2. 次式で与えられるベクトル $\boldsymbol{y}$ を求め，値を出力するプログラムを作成せよ．なお，行列 $\boldsymbol{A}$，ベクトル $\boldsymbol{x}$ の値は data 文で与えるものとする．

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix} \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{Bmatrix} \tag{1}$$

3. $\boldsymbol{x}_n$ が次式の漸化式で与えられるとき，$\boldsymbol{x}_5$ を求め，値を出力するプログラムを作成せよ．なお，行列 $\boldsymbol{A}$，ベクトル $\boldsymbol{x}_0$ の値は data 文で与えるものとする．

$$\boldsymbol{x}_n = \boldsymbol{A}\boldsymbol{x}_{n-1}, \qquad \boldsymbol{A} = \begin{bmatrix} -1 & 2 & -3 & -4 & 5 \\ -6 & 7 & -8 & -9 & 10 \\ -11 & 12 & -13 & -14 & 15 \\ -16 & 17 & -18 & -19 & 20 \\ -21 & 22 & -23 & -24 & 25 \end{bmatrix}, \quad \boldsymbol{x}_0 \begin{Bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{Bmatrix} \tag{2}$$

4. 次式で与えられる行列 $\boldsymbol{C}$ を求め，値を出力するプログラムを作成せよ．なお，行列 $\boldsymbol{A}$，行列 $\boldsymbol{B}$ の値は data 文で与えるものとする．

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix} \begin{bmatrix} -1 & 2 & -3 & -4 & 5 \\ -6 & 7 & -8 & -9 & 10 \\ -11 & 12 & -13 & -14 & 15 \\ -16 & 17 & -18 & -19 & 20 \\ -21 & 22 & -23 & -24 & 25 \end{bmatrix} \tag{3}$$