情報処理　（第13回）　(2017/7/10・茂木)

本日のテーマ

- ファイル入出力

演習の解答例

Q1. 九九の表 (kuku.f)

```
c23456
      write(6,5000)((i*j,j=1,9),i=1,9)
 5000 format(9i3) ! 横に 9 個，それぞれ整数 3 桁幅で出力
end
```

Q1（参考）. 九九の表・書式を直接記入 (kuku2.f)

```
c23456
      write(6,'(9i3)')((i*j,j=1,9),i=1,9) ! 書式を write 文に直接記入
      end
```

Q2. 行列とベクトルの積 (array3.f)

```
c23456
      program intp
      implicit none
      integer i, j
      real A(5,5), x(5), y(5) ! 配列の型宣言
      data A/ 1.0, 6.0, 11.0, 16.0, 21.0,
     & 2.0, 7.0, 12.0, 17.0, 22.0,
     & 3.0, 8.0, 13.0, 18.0, 23.0,
     & 4.0, 9.0, 14.0, 19.0, 24.0,
     & 5.0, 10.0, 15.0, 20.0, 25.0/
      data x/ 1.0, 2.0, 3.0, 4.0, 5.0/
      do i=1, 5
         y(i)=0.0
         do j=1, 5
            y(i)=y(i)+A(i,j)*x(j)
         enddo
      enddo
      write(6,7000)(y(j),j=1,5) ! 出力
 7000 format(f10.2)
      end
```

Q3. 漸化式の計算 (array32.f)

```
c23456
      program intp
      implicit none
      integer i, j, n
      real A(5,5), x(5), x_new(5) ! 配列の型宣言
      data A/-1.0, -6.0,-11.0,-16.0,-21.0,
     & 2.0, 7.0, 12.0, 17.0, 22.0,
     & -3.0, -8.0,-13.0,-18.0,-23.0,
     & 4.0, 9.0, 14.0, 19.0, 24.0,
     & -5.0,-10.0,-15.0,-20.0,-25.0/
      data x/ 1.0, 2.0, 3.0, 4.0, 5.0/
      do n=1, 6 ! {x1}〜{x6}まで繰り返す
         do i=1, 5
            x_new(i)=0.0
            do j=1, 5
               x_new(i)=x_new(i)+A(i,j)*x(j)
            enddo
         enddo
         do i=1, 5 ! 次のループに備え，x_new をx にコピー
            x(i)=x_new(i)
         enddo
      enddo
      write(6,7000)(x_new(j),j=1,5) ! 出力
 7000 format(f15.2)
end
```

Q.4 行列の掛け算 (arraym.f)

```
c23456
      program arraym
      integer i, j, k
      real a(5,5), b(5,5), c(5,5)   ! 二次元配列の型宣言
      data a/ 1.0,  6.0, 11.0, 16.0, 21.0,
     &        2.0,  7.0, 12.0, 17.0, 22.0,
     &        3.0,  8.0, 13.0, 18.0, 23.0,
     &        4.0,  9.0, 14.0, 19.0, 24.0,
     &        5.0, 10.0, 15.0, 20.0, 25.0/
      data b/-1.0, -6.0,-11.0,-16.0,-21.0,
     &        2.0,  7.0, 12.0, 17.0, 22.0,
     &       -3.0, -8.0,-13.0,-18.0,-23.0,
     &        4.0,  9.0, 14.0, 19.0, 24.0,
     &       -5.0,-10.0,-15.0,-20.0,-25.0/

      do i=1,5                        ! [c]=[a][b] の計算
         do j=1,5
            c(i,j)=0.0                ! 初期化のタイミング注意
            do k=1, 5
               c(i,j)=c(i,j)+a(i,k)*b(k,j)
            enddo
         enddo
      enddo
      do i=1, 5
         write(6,7000)(c(i,j),j=1,5)
      enddo
 7000 format(5f10.3)
      end
```

1 ファイル入出力

行列の積を計算・ファイル入出力版(arraym2.f)

```
 1:c23456
 2:       program arraym
 3:       implicit none
 4:       character*20 outfile !出力ファイル名を記憶する文字型変数 (最大 20 字)
 5:       integer i, j, k, m, n
 6:       parameter(m=10)  !行列の大きさを parameter 文で指定すると変更が楽
 7:       real A(m,m), B(m,m), C(m,m)   !二次元配列の型宣言（大きめに宣言）
 8:
 9:       open(10,file='in.txt',status='old',err=9999) !入力ファイルオープン
10:          !（装置番号 10，ファイルは既存，ファイルがないとき 9999 行へ）
11:       read(10,*) n              !配列の大きさを読込み，大きさをチェック
12:       if(n.gt.m) stop '入力した配列が大きすぎます．' ! 念のため大きさ確認
13:       do i=1, n
14:          read(10,*)(A(i,j),j=1,n)   !入力ファイルから行列 a を読込む
15:       enddo
16:       do i=1, n
17:          read(10,*)(B(i,j),j=1,n)   !引続き入力ファイルから行列 B を読込む
18:       enddo
19:       close(10)                     !読み終わったのでファイルクローズ
20:
21:       do i=1,n                      ! [c]=[a][b] の計算
22:          do j=1,n
23:             C(i,j)=0.0
24:             do k=1, n
25:                C(i,j)=C(i,j)+A(i,k)*B(k,j)
26:             enddo
27:          enddo
28:       enddo
29:
30:       outfile='out.csv'
31:       open(12,file=outfile) !出力ファイルオープン（装置番号 12 とする）
32:       do i=1, n
33:          write(12,7000)(C(i,j),j=1,n)     !出力ファイルに書き出し
34:       enddo
35:       close(12)                         !出力ファイルをクローズ
36: 7000 format(10(f10.3,:','))
37:       write(6,*)'結果をファイル out.csv に出力しました．'
38:       stop
40: 9999 write(0,*)'in.txt のオープンに失敗しました'
41:       end
```

入力ファイル in.txt の内容

```
5                                  ←行列の大きさ
  1.0,  2.0,   3.0,   4.0,  5.0    ←行列 A
  6.0,  7.0,   8.0,   9.0, 10.0
 11.0, 12.0,  13.0,  14.0, 15.0
 16.0, 17.0,  18.0,  19.0, 20.0
 21.0, 22.0,  23.0,  24.0, 25.0
 -1.0,  2.0,  -3.0,  -4.0,  5.0   ←行列 B
 -6.0,  7.0,  -8.0,  -9.0, 10.0
-11.0, 12.0, -13.0, -14.0, 15.0
-16.0, 17.0, -18.0, -19.0, 20.0
-21.0, 22.0, -23.0, -24.0, 25.0
```

2 OPEN 文

これまでの講義では，データの入力(read)，出力（write）はそれぞれキーボード（装置番号 5），画面（装置番号 6）で行っていましたが，任意のファイルから直接，入出力することができます．このためには，入出力を行いたいファイルに対して OPEN 文を使って装置番号を割り当てます．OPEN 文は次のように書きます．（詳細は教科書 p.106）．

open(装置番号,file=ファイル名,status=既存・新ファイル指定,err=オープン失敗時のエラー処理行番号)

- **装置番号**はすでに 5（キーボード），6（画面・標準出力），0（画面・標準エラー出力）が割り当て済みですので，これ以外の任意の番号（正の整数値・２桁が一般的）を用います．

- **ファイル名**はプログラム例の 9 行目 open(10,file='in.txt',...) のように，引用符でくくった文字定数を与えることも，30 行目 open(12, file=outfile) のように，文字型変数で与えることもできます．例えば，
  open(10,file='in.txt') と指定すると，この open 文以降は in.txt の装置番号として 10 が割り当てられ，このファイルからの入力が read(10,*)...，このファイルへの出力が write(10,*)... でできるようになります．

- **ファイル属性** (status) は，指定したファイルが既存のファイルの時 status='old'，新たにファイルを作成するとき status='new' とします．status は省略可能ですので，入力用のファイルに対してのみ status='old' を指定し，出力用のファイルには何も指定しないのが使いやすいと思います．

- **エラー指定子** (err) は，オープンに失敗したとき（例えば status='old' でオープンする場合に，指定したファイル名のファイルが存在しない場合や，status='new' でオープンする場合に，すでに指定したファイル名のファイルが存在する場合など）に，エラー処理を行う部分の行番号を指定します．通常はプログラム例の 38 行目のように，エラーメッセージを表示してプログラムを終了させてしまうのが簡単です．この指定子も省略可能です．

3 CLOSE 文

OPEN 文でファイルをオープンし必要な入出力を完了したら CLOSE 文を実行してファイルをクローズしておきます．なお，CLOSE 文を実行しない場合，プログラムが終了するときに自動的にクローズされます．一度クローズすると割り当てられた装置番号が無効になり，そのプログラムからアクセスできなくなるため，誤って貴重なファイルを損傷する危険がなくなります．書き方は単に，
close(装置番号)
とします．プログラム例では，19 行目，34 行目で CLOSE 文を使っています．

練習問題

1. 二つの 0〜1 の一様乱数を x，y とするとき，次式で与えられる s，t は，互いに独立な，平均値 μ=0，標準偏差 σ=1 の正規乱数（正規分布に従う乱数）となる（ボックス・ミューラー法）.

$$\begin{cases} s = \sqrt{-2\log_e x}\,\sin(2\pi y) \\ t = \sqrt{-2\log_e x}\,\cos(2\pi y) \end{cases} \tag{1}$$

   上記の方法で 5000 個の正規乱数（すなわち s，t が 2500 組）を発生させて，ファイル `Gauss.csv` に書き出すプログラムを作成せよ.

2. 1. で作成した `Gauss.csv` を読み込んで，-3〜3 の範囲の発生頻度を 0.2 刻みでカウントし，下の例のように階級値（中央の値）, 発生確率を `Probability.csv` に出力するプログラムを作成せよ.

Probability.csv の内容

```
-2.90,    0.0016000  ←-3.0〜-2.8 の階級値　,　 発生確率
-2.70,    0.0028000
-2.50,    0.0030000  ←カンマ区切り，ファイル名を「名前.csv」にしておく
      略                  と Excel でグラフが簡単に描けます.
 2.90,    0.0010000
```

3. 次の上三角行列（対角成分より下の成分がすべて 0 で，対角成分が 0 でない行列）を係数行列とする連立一次方程式を解くプログラムを作成せよ．なお，繰り返し計算には `do` ループを用いるものとする.

$$\begin{bmatrix} 2 & -1 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 2 \\ -2 \end{Bmatrix}$$