

本日のテーマ

- DO ループ

1 先週の演習の解答例

Q1 階乗の計算

```
c23456 10の階乗
      program factorial
      implicit none
      integer i, n, k

      i=10
      k=1
      n=2
1000   k=k*n
c      write(6,*)n, k ! ループの動作チェック用・チェックしたい
c      read(5,*)      ! ときは先頭のcを取ってコンパイル
      if(n.eq.i) goto 2000
      n=n+1
      goto 1000
2000 write(6,*)i, '!=', k, 'です. '

      end
```

Q2 平方根

```
c23456
      program square_root
      implicit none
      real a, b, aa

      write(6,*)'bを入力してください. '
      read(5,*)b

      a=1
1000 aa=0.5*(b/a+a)
      write(6,*) aa
      if( abs(aa-a) .gt. 1.0E-6 ) then
          a=aa
          goto 1000
      endif

      end
```

```

c23456
  program logx
  real x, ans, a
  integer i

  write(6,*) 'x の値を入力 (x>0)'
  read(5,*)x
  ans=0.0
  i=1
1000  a=((x-1.0)/(x+1))**(2*i-1)/(2*i-1)*2.0
      ans=ans+a
      write(6,*)i, a, ans
      if(abs(a).lt.1.0E-9) goto 2000
      i=i+1
      goto 1000
2000  stop
      end

```

```

c23456
  program co
  integer m, mx

  write(6,*) '数値 m を入力して下さい. '
  read(5,*)m

  mx=0      ! 最大値の記録用
1000  if( m.eq.1 ) goto 2000      ! ループの脱出条件
      if( m .eq. m/2*2 ) then    ! m は偶数?
          m=m/2                  ! 偶数のとき
      else
          m=m*3+1                ! 奇数のとき
      endif
      if(mx.lt.m) mx=m
      write(6,*)m
      goto 1000

2000  write(6,*) 'm の最大値は',mx,' です. '
      end

```

2 DO ループ

do ループはループの初期値と終了値を与えてループを構成するもので、goto 文と if 文を組み合わせたループよりもプログラムが見やすくなります。

```
c23456
do 文番号 DO変数 = 初期値, 終了値(, 増分)
.....
.. 繰り返し実行される命令群 (DO ブロック) ...
.....
文番号 continue
```

また、文番号を使わない次のタイプもあります。

```
c23456
do DO変数 = 初期値, 終了値(, 増分)
.....
.. 繰り返し実行される命令群 (DO ブロック) ...
.....
enddo
```

増分は、DO 変数が整数型でかつ増分が1のとき省略できます。DO 変数には実数型、倍精度実数型も使うことができますが、この場合には増分は省略できません。なお、DO 変数には実数型、倍精度実数型を使うと計算に誤差が含まれるため終了条件が問題になることがあります（将来廃止されるようですので使わない方がよいでしょう）。また、「**繰り返し実行される命令群**」中で **do 変数の値を書き換えてはいけません**。

2.1 DO ループの流れ

次のサンプルプログラムを実行して、DO ループの実行順序を確かめてみましょう。このプログラムの do 変数は n、初期値は 2、終了値は i=4、増分は 1 です。

DO ループの動作確認プログラム (factorial2.f · 左の数字とコロンは無視)

```

1:c23456 4の階乗
2:      program factorial2
3:      implicit none
4:      integer i, n, k
5:
6:      i=4
7:      k=1
8:
9:      do n=2, i, 1
10:         k=k*n
11:      enddo
12:
13:      write(6,*)i,'!','=',k,'です. '
14:
15:      write(6,*)'プログラムを終了します. エンターキーを押してください. '
16:      read(5,*)
17:      end

```

プログラムの実行順序とそれに伴う変数の値の変化をまとめると次ページのようにになります。

実行される行	変数の値と変化		(説明)・"画面の表示"
	n	k	
7: k=1	不定	不定 → 1	
9: do n=2, i, 1	不 → 2	1	(DO ループに入る.)
10: k=k*n	2	1 → 1×2=2	
11: enddo	2	2	(無条件に DO 文(9行目)に戻る)
9: do n=2, i, 1	2 → 3	2	(DO ループ 2 巡目.)
10: k=k*n	3	2 → 2×3=6	
11: enddo	3	6	(無条件に DO 文(9行目)に戻る)
9: do n=2, i, 1	3 → 4	6	(DO ループ 3 巡目.)
10: k=k*n	4	6 → 6×4=24	
11: enddo	4	24	(無条件に DO 文(9行目)に戻る)
9: do n=2, i, 1	4 → 5	24	(ループ変数 n(=5) が終了値 4 を超えたのでループを抜ける.)
13: write(6,*) i,'!','=', ...	5	24	" 4! = 24 です. "
15: write(6,*) 'プログ...	5	24	(いつもの終了メッセージ)

練習・宿題

1. DO ループを用いて,

$$1 + 3 + 6$$

を求めるプログラムを作成し、左ページと同様の変数の変化表を作成せよ。

2. 2~100 までの素数をすべて出力するプログラムを作成せよ。素数の判定は、判定したい数 n に対して $2, \dots, n/2$ で順に割って余りを調べればよい。
3. 2~500 までの素数の和を求めるプログラムを作成せよ。
4. 下の例を参考に 0~1 の一様乱数を 1000 個生成し、生成した乱数の平均値を求めよ。

0~1 の一様乱数を 50 個発生させ画面に出力するプログラム例

```
c23456
  program Monte_Carlo
    integer i, n
    real x

    n=50
    do i=1, n
      x = rand() ! rand (非標準の組込関数) が呼ばれると 0~1 の一様乱数
        を 1 つ発生する.
      write(6,*) x
    enddo
  end
```