

本日のテーマ

- GOTO 文

1 先週の演習の解答例

Q1. ヘロンの公式 (改良版)

```
c23456
program menseki3
implicit none
real area, a, b, c

write(6,*) '三辺の長さを入力して下さい. '
read(5,*) a, b, c

s=(a+b+c)/2.0
s=s*(s-a)*(s-b)*(s-c)

if(s.lt.0.0) then ! s<0 か判定
  write(6,*) '三角形になりません. ' ! s<0 のときメッセージを表示するだけ
else
  area=SQRT(s) ! s>=0 を確認してから平方根を計算していることに注意
  write(6,*) '面積は', area, 'です. '
endif
end
```

Q2. 角度の計算 ($x=0$ のとき難あり)

```
c23456 角度の計算
program kakudo2
implicit none
real x, y, theta, pi
pi=3.1415927
write(6,*) 'x座標とy座標の値を入力してください. '
read(5,*) x, y
theta=atan(abs(y)/abs(x)) ! 組込関数 ATAN は rad 単位の角度を返す
if( x.ge.0 ) then
  if( y.ge.0 ) then
    write(6,*) '第1象限 角度は', theta*180/pi, '度'
  else
    write(6,*) '第4象限 角度は', 360.0-theta*180/pi, '度'
  endif
else
  if( y.ge.0 ) then
    write(6,*) '第2象限 角度は', 180.0-theta*180/pi, '度'
  else
    write(6,*) '第3象限 角度は', theta*180/pi+180.0, '度'
  endif
endif
end
```

Q2. 角度の計算 (改良版)

c23456 角度の計算 (x=0 のときの問題対応後)

```

program kakudo3
implicit none
real x, y, theta, pi
pi=3.1415927
write(6,*)'x 座標と y 座標の値を入力してください. '
read(5,*)x, y

if( x.gt.0 .and. y.ge.0 ) then
  write(6,*)' 第1象限 角度は', atan(y/x)*180/pi,' 度'
elseif( x.le.0.0 .and. y.gt.0.0 ) then
  write(6,*)' 第2象限 角度は', atan(x/(-y))*180/pi+90.0,' 度'
elseif(x.lt.0.0 .and. y.le.0.0 ) then
  write(6,*)' 第3象限 角度は', atan(y/x)*180/pi+180.0,' 度'
elseif(x.ge.0.0 .and. y.lt.0.0 ) then
  write(6,*)' 第3象限 角度は', atan(x/(-y))*180/pi+270.0,' 度'
else
  write(6,*)' 原点です. '
endif
end

```

Q3. 西暦の計算

c23456 明治 1868~, 大正 1912~, 昭和 1926~, 平成 1989~

```

program seireki
integer nen

write(6,*)' 西暦入力'
1000 read(5,*)nen
if(nen.lt.1868) then
  write(6,*)' 明治元年 (1868 年) 以降を入力して下さい'
  goto 1000
endif
if( nen.ge.1989 ) then
  write(6,*)' 平成',nen-1989+1,' 年'
elseif( nen.ge.1926) then
  write(6,*)' 昭和',nen-1926+1,' 年'
elseif( nen.ge.1912) then
  write(6,*)' 大正',nen-1912+1,' 年'
else
  write(6,*)' 明治',nen-1868+1,' 年'
endif
end

```

```

c23456
program niji
implicit none
real a, b, c, x1, x2, D

write(6,*) '係数 a, b, c を入力して下さい. '
read(5,*) a, b, c
D=b*b-4.0*a*c ! いきなり解を求めずに, まず判別式を計算
if(D.ge.0.0) then ! D>=0を確認して
  x1 = (-b-SQRT(D))/2.0/a ! SQRT (平方根) を求めている
  x2 = (-b+SQRT(D))/2.0/a
  write(6,*) '実数解', x1, 'と', x2, 'です. '
else ! D<0 のときは複素解なので実部と虚部を別に求める
  x1 = -b/2.0/a ! 実部
  x2 = SQRT(-D)/2.0/a ! 虚部 (このとき -D>0 に注意)
  write(6,*) '複素解', x1, ' ± ', x2, 'i です. '
endif
stop
end

```

2 GOTO文

2.1 goto文の動作

goto文が実行されると、goto文に付けられた文番号(statement label・下のプログラム例では1000あるいは2000)の文に無条件に制御が移されます(ジャンプするという)。文番号は、1~5カラム目に書いておく任意の番号で、プログラムに同じ番号が他にあってはいけません。

通常、プログラムは上の行から順に実行されていきますが、goto文を使うとプログラムの流れを変えることができます。このような文を制御文と呼びます。

2.2 goto文を用いたループ

実際のプログラミングでは繰り返し計算(ループ・loop)が多用されます。これは決まった処理を繰り返し行うもので、コンピュータの真価が発揮される場面です。ループの制御文としてFORTRANではdo文がありますが、goto文とif文を組み合わせることでループを構成することもできます。goto文を用いたループは、繰り返し回数があらかじめわからない場合にはよく用いられます。

次の例は、1~3まで順にたし合わせる(1+2+3)プログラムです。

ループの例 (sum.f)

```
c23456
  program sum
  implicit none
  integer n, i, nsum

  n=3                ! 3まで足しあわせるものとする
  nsum=0            ! nsumに何が入っているかわからないので0で初期化
  i=1              ! nの初期値を1にする
1000 nsum=nsum+i
      if(i.eq.n) goto 2000    ! iがn (=3) になったら足し合わせを終了
      i=i+1                  ! iを1増やす
      goto 1000

2000 write(6,*) '1 から', n, ' までたすと', nsum, ' になります. '
  end
```

練習・宿題

1. sum.f を参考に、10 の階乗を計算・出力するプログラムを作成せよ。
2. 次の漸化式で与えられる a_i は b の平方根 \sqrt{b} に収束する。 b を入力し、この漸化式で与えられる a_i の値を、 $|a_i - a_{i-1}| < 1.0 \times 10^{-6}$ となるまで順に計算・出力するプログラムを作成せよ。

$$a_i = \frac{1}{2} \left(\frac{b}{a_{i-1}} + a_{i-1} \right), \quad a_0 = 1$$

ヒント: 1.0×10^{-7} をプログラム中に記す場合は、 $1.0E-7$ と書く。また、収束の判定は $a_i - a_{i-1}$ が負値になることを想定し、二乗して比較、もしくは、絶対値関数 ABS を用いて比較する。

3. $\log_e x$ の値は次式の漸化式で評価することができる。 x の値を入力させ、次式に従って $\log_e x$ ($x > 0$) の値を計算・出力するプログラムを作成せよ。

$$\frac{\log_e x}{2} = \left(\frac{x-1}{x+1} \right) + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \cdots + \frac{1}{2n-1} \left(\frac{x-1}{x+1} \right)^{(2n-1)}$$

なお、

$$\frac{1}{2n-1} \left(\frac{x-1}{x+1} \right)^{(2n-1)} < 1.0 \times 10^{-7}$$

となったときに計算を打ち切るものとする。

4. 正の整数 n に対し、(1) n が偶数の時は2で割る、(2) n が奇数の時は3倍し、1を足す、という操作を繰り返していくうちに結果が1になる (コラッツの予想)。
任意の整数を入力させて上記の計算を行い、1になるまでの値を順に示すプログラムを作成し、作成したプログラムと自分の学籍番号の数字部分 (例えば 03TC091 だったら 3091) を入力したときに、計算の途中に現れる最も大きい値を記せ (余力のある人は、最大値を記憶しておく部分をプログラムに付加するとよい)。