2019年度前期　金曜4限

情報処理演習

第5回

2019/05/31

平山　修久

名古屋大学減災連携研究センター
Disaster Mitigation Research Center, NAGOYA UNIVERSITY

NAGOYA UNIVERSITY

前回までの復習

Step 1　プログラムを書く

- エディタ（メモ帳）でソースコードを入力し，ファイル「***.f90**」（拡張子）で保存。

Step 2　プログラムをコンパイルする

- コマンドプロンプトで，「**gfortran プログラム名**」で，「**a.exe**」の実行ファイルを生成。

Step 3　プログラムを実行

- コマンドプロンプトで，「**a.exe**」と入力してEnter

単純論理式

- 論理定数（.TRUE.と.FALSE.）

式1　関係演算子　式2

演算子		意味
<	または .LT.	より小さい
>	または .GT.	より大きい
.EQ.	または ==	等しい
<=	または .LE.	以下
>=	または .GE.	以上
/=	または .NE.	等しくない

NAGOYA UNIVERSITY

複合論理式

- .NOT.（否定），.AND.（論理積），.OR.（論理和），.EQV.（等価），.NEQV.（排他的論理和）

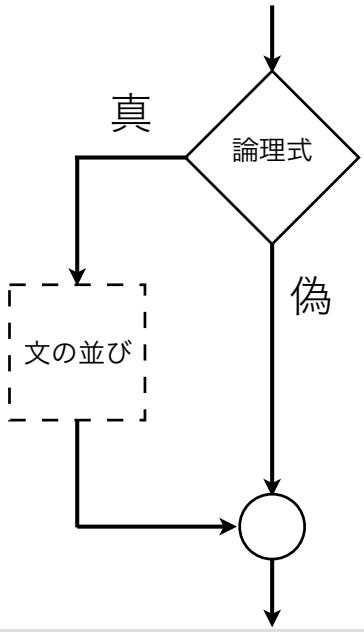
p	.NOT. p
.TRUE.	.FALSE.
.FALSE.	.TRUE.

p	q	p .AND. q	p .OR. q	p .EQV. q	p .NEQV. q
.TRUE.	.TRUE.	.TRUE.	.TRUE.	.FALSE.	.TRUE.
.TRUE.	.FALSE.	.FALSE.	.TRUE.	.FALSE.	.TRUE.
.FALSE.	.TRUE.	.FALSE.	.TRUE.	.FALSE.	.TURE.
.FALSE.	.FALSE.	.FALSE.	.FALSE.	.TRUE.	.FALSE.

IF構文

- 最も単純な選択構造

```
IF (論理式) THEN
    文の並び
END IF
```




```
IF (X >= 0) THEN
    Y = X * X
    Z = SQRT(X)
END IF
IF (1.5 <= X .AND. x <= 2.5) PRINT *,X
```

NAGOYA UNIVERSITY

IF-ELSE IF構文

- 3つ以上の選択肢が含まれるとき

```
IF (論理式1) THEN
   文の並び1
ELSE
   IF (論理式2) THEN
      文の並び2
   ELSE
      文の並び3
   END IF
END IF
```

- IF-ELSE IF構文を使うと

```
IF (論理式1) THEN
   文の並び1
ELSE IF (論理式2) THEN
   文の並び2
ELSE IF (論理式3) THEN
   文の並び3
ELSE
   文の並びn
END IF
```

CASE構文

```
SELECT CASE (場合式)
    CASE (場合値リスト1)
        文の並び1
    CASE (場合値リスト2)
        文の並び2
    CASE (場合値リスト3)
        文の並び3
END SELECT
```

- 場合式は，整数式，文字式，論理式
- 場合値リストは，場合式が取りうる1つ以上の値を括弧で囲んだリストか，キーワードDEFAULT
- (値)【単一の値】，(値1:値2)【値1から値2までの範囲】，(値1:)【値1以上のすべての値の集合】，(:値2)【値2以下のすべての値の集合】

NAGOYA UNIVERSITY

繰り返し実行：カウンタ制御DOループ

```
DO 制御変数 = 初期値, 限界値, 増分値
    文の並び
END DO
```

1.制御変数に初期値が代入される。

2.制御変数が限界値と比較され，次の条件を満たすかどうか調べる
- 増分値が正の場合は，限界値以下であるかどうか
- 増分値が負の場合は，限界値以上であるかどうか

3.条件を満たす場合は，ループ本体と呼ばれる文の並びが実行され，制御変数に増分値が加算されて，ステップ2が繰り返される。そうでない場合は繰り返しが終了する。

繰り返し実行：汎用DOループ(1)

> Do-EXIT構文

```
DO
    文の並び1
    IF (論理式) EXIT
    文の並び2
END DO
```

1.文の並びに1または文の並び2は省略できる。

2.ループ本体を構成する文は，IF文の論理式が真になるまで繰り返される。IF文の論理式が真になると，その時点で繰り返しは終了し，END DOの次の文に実行が続く。

3.論理式が決して真にならない場合は，**無限ループ**となる。

NAGOYA UNIVERSITY

繰り返し実行：汎用DOループ(5)

> EXIT文は，END DO文の次の文に制御を移すことによってループの繰り返しを終了させる。

> 現在の繰り返しだけを途中で終了させる：**Do-CYCLE構文**

```
DO
    READ *, Celsius
    IF (Celsius < 0.0) THEN
        PRINT *, "/// Temperature must be 0 or above"
        CYCLE
    END IF
    文の並び
END DO
```

1.Celsiusに負の値が入力されると，メッセージが表示され，残りのループ文をスキップ。ループの繰り返しが新たに開始される。

DOループの入れ子

```
  DO M = 1, Last_M
      DO N = 1, Last_N
          Product = M * N
          PRINT *, M, " ", N, " ", Product
      END DO
  END DO
```

```
MacPro2017:0531 nhirayama$ gfortran demo5_1.f90
MacPro2017:0531 nhirayama$ ./a.out
           1           1           1
           1           2           2
           1           3           3
           2           1           2
           2           2           4
           2           3           6
           3           1           3
           3           2           6
           3           3           9
           4           1           4
           4           2           8
           4           3          12
MacPro2017:0531 nhirayama$
```

!DO変数の変更は不可

```
DO I = 1, 200
    I = 100
END DO
```


数の表現と誤差

- 0.1を10回足しても1.0にならない！
- コンピューターでは，2進数で表現する。

  0.1 = (0.CCCCD)16×2-3

```
IMPLICIT NONE
REAL :: N
INTEGER :: I
N=0.0
DO I = 1, 10
        N = N + 0.1
        PRINT *, N
END DO
```

```
MacPro2017:0531 nhirayama$ ./a.out
  0.100000001
  0.200000003
  0.300000012
  0.400000006
  0.500000000
  0.600000024
  0.700000048
  0.800000072
  0.900000095
   1.00000012
MacPro2017:0531 nhirayama$
```

単精度実数ではなく，倍精度実数とする

単精度実数

```
IMPLICIT NONE
REAL :: N
INTEGER :: I
N=0.0
DO I = 1, 10
        N = N + 0.1
        PRINT *, N
END DO
```

倍精度実数

```
IMPLICIT NONE
DOUBLE PRECISION :: N
INTEGER :: I
N=0.0d0
DO I = 1, 10
        N = N + 0.1d0
        PRINT *, N
END DO
```

```
MacPro2017:0531 nhirayama$ gfortran demo5_1.f90
MacPro2017:0531 nhirayama$ ./a.out
  0.100000001
  0.200000003
  0.300000012
  0.400000006
  0.500000000
  0.600000024
  0.700000048
  0.800000072
  0.900000095
   1.00000012
MacPro2017:0531 nhirayama$ gfortran demo5_1.f90
MacPro2017:0531 nhirayama$ ./a.out
  0.10000000000000001
  0.20000000000000001
  0.30000000000000004
  0.40000000000000002
  0.50000000000000000
  0.59999999999999998
  0.69999999999999996
  0.79999999999999993
  0.89999999999999991
  0.99999999999999989
MacPro2017:0531 nhirayama$
```

単精度実数
有効数字6桁

倍精度実数
有効数字15桁

桁落ち(1)

- 絶対値がごく近い2数を足したり引いたりして結果の絶対値が小さくなるような計算をすると，絶対値が小さくなった分だけ相対誤差が大きくなる：
  有効数字が減る＝桁落ち

$$1-\frac{1}{\sqrt{1+x}}=\frac{\sqrt{1+x}-1}{\sqrt{1+x}}=\frac{x}{(1+x)+\sqrt{1+x}}$$

```
X = 0.0031834

PRINT *, 1.0 - 1.0 / SQRT(1.0 + X)
PRINT *, (SQRT(1.0 + X) - 1.0) / SQRT(1.0 + X)
PRINT *, X / ((1.0 + X) + SQRT(1.0 + X))
```

```
MacPro2017:0531 nhirayama$ ./a.ou
   1.58786774E-03
   1.58784585E-03
   1.58791000E-03
MacPro2017:0531 nhirayama$
```

0.00158791 = 1.58791E-03

桁落ち(2)

二次方程式（$ax^2+bx+c=0$）の解の公式

$4ac$が0に小さい場合や判別式Dが0に近い場合

$$2.718282\,x^2 - 684.4566\,x + 0.3161592 = 0$$

$$2.718282\,x^2 - 1.854089\,x + 0.3161592 = 0$$

NAGOYA UNIVERSITY

演習課題5

＞ 提出物：３つ

1.kadai5_1.f90（プログラム）

2.kadai5_1.png（画像ファイル）

3.kadai5_2.f90（プログラム）

桁落ち(2)

二次方程式（$ax^2+bx+c=0$）の解の公式

$4ac$が0に小さい場合や判別式Dが0に近い場合

$2.718282\,x^2 - 684.4566\,x + 0.3161592 = 0$

251.797, 4.61914E-4

```
[MacPro2017:0510 nhirayama$ ./kadai3_1.out
 Enter the coefficients of A, B, C:
2.718282 -684.4566 0.3161592
 The roots are    251.797028        4.60298354E-04
```

$2.718282\,x^2 - 1.854089\,x + 0.3161592 = 0$

0.341512, 0.340569

```
[MacPro2017:0510 nhirayama$ ./kadai3_1.out
 Enter the coefficients of A, B, C:
2.718282 -1.854089 0.3161592
 The roots are  0.341524273      0.340556920
```

倍精度実数で計算すると

```
[MacPro2017:0510 nhirayama$ ./a.out
 Enter the coefficients of A, B, C:
2.718282 -684.4566 0.3161592
 The roots are    251.79703371052130
                  4.6191355300849733E-004
```

```
[MacPro2017:0510 nhirayama$ ./a.out
 Enter the coefficients of A, B, C:
2.718282 -1.854089 0.3161592
 The roots are  0.34151206351711483
                0.34056911864132211
```

NAGOYA UNIVERSITY

課題5_2　ニュートン法

- **倍精度実数（DOUBLE PRECISION）を用いること。**
- **繰り返しを終了する誤差の許容値を決めておく，er0=1.0d-6**
- **アルゴリズム**

1.aを入力する。

2.aが負の場合には，プログラムを止める（STOP文）

3.まずx1にaを代入する。

4.1)～4)を，1～KM(=100)まで繰り返す。

1)x2 = x1 - 0.5d0 * (x1**2 - a) / x1 を計算する。

2)誤差erを計算する。絶対値ABS。

3)誤差erが誤差許容値er0以下ならDOループを終了する。

4)x1にx2に代入する。

5.x2，K（繰り返し回数），er（誤差）を表示する。