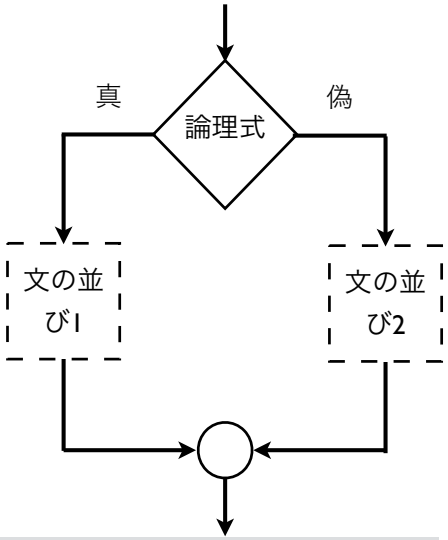
2025年度前期
情報処理演習
第5回
2025.5.21

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

IF構文
汎用形式

```
IF (論理式) THEN
   文の並び1  論理式が真（.TRUE.）
ELSE
   文の並び2  論理式が偽（.FALSE.）
END IF
```




```
IF (x>0) THEN
   print *, “x is positive.”
ELSE
   print *, “x is nagative.”
END IF
```

条件分岐（IF，CASE）

成績の表示

```
IF (N >= 90) THEN
    PRINT *, “S”
ELSE IF (N >= 80) THEN
    PRINT *, “A”
ELSE IF (N >= 70) THEN
    PRINT *, “B”
ELSE IF (N >= 60) THEN
    PRINT *, “C”
ELSE
    PRINT *, “F”
END IF
```

```
SELECT CASE (N)
CASE (90:100)
    PRINT *, “S”
CASE (80:89)
    PRINT *, “A”
CASE (70:79)
    PRINT *, “B”
CASE (60:69)
    PRINT *, “C”
CASE DEFAULT
    PRINT *, “F”
END SELECT
```


Tips

前回課題のヒント

- **条件分岐：**研究においては，想定していない値となった場合について考えることが必要。0での除算や想定した入力データ以外の場合。

```
/2*a　と　/(2*a)　と　/2/a　の違い
```

- **CASE文の場合**には，最後にDEFAULTを入れることが，思わぬ計算違いを防ぐ。場合式は整数（INTEGER）

```
SELECT CASE (場合式) → INT(Amount)
    CASE (場合値リスト1)
        文の並び1
    DEFAULT
        文の並び3
END SELECT
```

本日の目標

2025.5.21

- 繰り返しの計算ができる。
  - フィボナッチ数を計算できる
  - 定積分を計算できる

繰り返し実行

カウンタ制御DOループ

```
DO i = 1, 10
    文の並びi*i
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
   1. 増分値が正の場合は，限界値以下であるかどうか
   2. 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

カウンタ制御DOループ

```
MacPro2017:0517 nhirayama$ ./a.out
           1           1
           2           4
           3           9
           4          16
           5          25
           6          36
           7          49
           8          64
           9          81
```

```
DO Number = 1, 9
    PRINT *, Number, Number**2
END DO
```

- Numberは整数型変数，Numberが制御変数，初期値1，限界値9，増分値1
- 初期値，限界値，増分値には変数や式も指定できる。

```
READ *, Number
DO I = 1, Number
    Sum = Sum + I
END DO
```

- **計算結果：1+2+3+ … + Number**

繰り返し実行

汎用DOループ

＞ 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。

繰り返し実行

汎用DOループ：事前，事後

- 事前テストループ

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

- 事後テストループ

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


繰り返し実行

汎用DOループ例

- Limitが与えられた時，1+…+NumberがLimitより大きくなる正の整数値Numberの最小値を求める。

```
Number = 0
Sum = 0
DO
    IF (Sum > Limit) EXIT
    Number = Number + 1
    Sum = Sum + Number
END DO
PRINT *, Number
```

繰り返し実行

汎用DOループ：問い合わせ制御

> 問い合わせ制御入力ループ

```
DO
  PRINT *, “Enter temperature in degrees
Celsius :”
  READ *, Celsius
  Fahrenheit = 1.8 * Celsius + 32.0
  PRINT *, Celsius, “(C) = “, Fahrenheit, “(F)”
  PRINT *, “More temperatures to convert(Y or N)?”
  READ *, Response
  IF (Response == “N”) EXIT
END DO
```


繰り返し実行

問い合わせ制御入力ループ実行例

demo4_2

```
PROGRAM Demo4_2

        IMPLICIT NONE
        REAL :: Celsius, Fahrenheit
        CHARACTER(1) :: Response

        DO
                PRINT *, "Enter temperature in degrees Cel
                READ *, Celsius
                Fahrenheit = 9./5.*Celsius + 32.0
                PRINT *, Celsius, "degrees Celsius = ", Fa
                PRINT *
                PRINT *,"More temperatures to convert (Y o
                READ *, Response
                IF (Response == "N") EXIT
        END DO

END PROGRAM Demo4_2
```

0517 — -bash — 128×72

```
MacPro2017:0517 nhirayama$ gfortran demo4_2.f90
MacPro2017:0517 nhirayama$ ./a.out
 Enter temperature in degrees Celsius:
0
   0.00000000     degrees Celsius =    32.0000000     degreed Fahrenheit.

 More temperatures to convert (Y or N)?
Y
 Enter temperature in degrees Celsius:
100
   100.000000     degrees Celsius =    212.000000     degreed Fahrenheit.

 More temperatures to convert (Y or N)?
u
 Enter temperature in degrees Celsius:
80
   80.0000000     degrees Celsius =    176.000000     degreed Fahrenheit.

 More temperatures to convert (Y or N)?
y
 Enter temperature in degrees Celsius:
-100
  -100.000000     degrees Celsius =   -148.000000     degreed Fahrenheit.

 More temperatures to convert (Y or N)?
n
 Enter temperature in degrees Celsius:
800
   800.000000     degrees Celsius =    1472.00000     degreed Fahrenheit.

 More temperatures to convert (Y or N)?
N
MacPro2017:0517 nhirayama$
```

繰り返し実行

汎用DOループ

- **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。

今日の課題

25.5.21

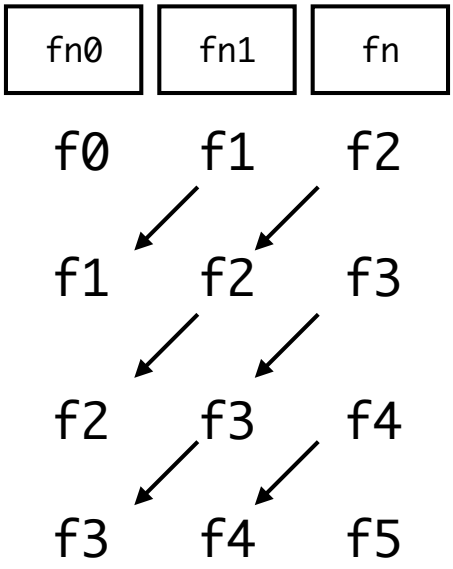
1. フィボナッチ数例$F_n = F_{n-1} + F_{n-2},\ \ F_0 = 0,\ \ F_1 = 1$を用いて，n番目（n>2）のフィボナッチ数，ならびに黄金比F_n/F_{n-1}の値を求めるプログラムを作成する。$n = 10,\ \ n = 25$のときのフィボナッチ数，黄金比の値を計算する。**提出物はプログラム（kadai5_1.f90）**

2. 台形公式$S = \frac{1}{2}\Delta x \sum_{k=0}^{n-1} (y_k + y_{k+1}) = \left(\frac{1}{2}(y_0 + y_n) + y_1 + y_2 \ldots + y_{n-1}\right)\Delta x$を用いて，$\int_0^1 x(x-1)(x-2)dx$の定積分の近似値を求めるプログラムを作成する。$n = 100,\ \ n = 1{,}000,\ \ n = 10{,}000$の時の定積分の近似解をそれぞれ求めよ。**提出物は実行結果（kadai5_2.png）**

課題5_1

フィボナッチ数列のアルゴリズム

1. nを入力する。
2. 繰り返し計算（i=2からn）
    1) fn=fn0+fn1を計算する。
    2) fn0にfn1を代入する。
    3) fn1にfnを代入する。
3. 黄金比golden_ratioを計算する。
4. fnとgolden_ratioを表示する。



課題5_2

数値積分のアルゴリズム

1. nを入力する。
2. 分割幅dxを計算する。
3. 面積sを初期化する。
4. 繰り返し（i=0からn）計算
    1) x軸上のi番目の値xを計算する。
    2) 値xに対応するyを計算する。
    3) 両端の点のとき，i=0，i=nのとき，yを0.5倍する。
    4) 面積sにyを加算する。
5. 面積sにdxを乗じて近似値を計算する。
6. 面積sを表示する。