

本日の授業のテーマ

今までの学習内容をまとめます。

- (1) プログラムの流れと制御文
- (2) IF 文
- (3) 論理式を構成する関係演算子
- (3) GO TO 文
- (4) ループ構造
- (5) 初期化とカウンター
- (6) 足し合わせ
- (7) 教科書の練習問題について
- (8) 実際のプログラム
 - ・和を求めるプログラム

本日の授業のゴールは、以下のとおり。

- ・FORTRAN のプログラムは、実行文が上から下に 1 行ずつ処理される。ただし、制御文 (IF 文や GO TO 文など) があると処理の順序が変更される。プログラムの実行の流れが分かる。
- ・ブロック IF 文、論理 IF 文、算術 IF 文の使い方が分かる。
- ・論理式の意味がわかり、関係演算子の使い方が分かる。
- ・GO TO 文があった場合の処理の流れが分かる。
- ・ループ構造が分かり、GO TO 文と IF 文でそのプログラムが書ける。
- ・初期化の意味がわかり、カウンターの動作が理解できる。
- ・足し合わせの方法が理解できる。
- ・教科書の演習問題が理解できる。
- ・IF 文を使ったプログラムが書ける。

1 プログラムの流れと制御文

- FORTRAN のプログラムは、実行文が上から下へと 1 行ずつ処理されます (図 1)。

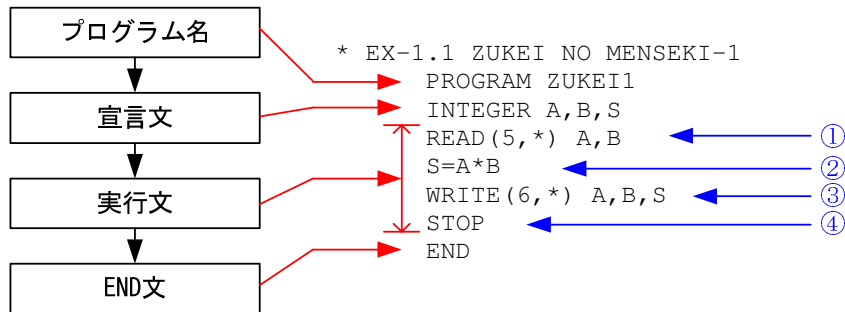


図 1 このプログラムは、①②③④と処理されます。

- しかし、IF 文や GO TO 文などがあれば、その処理の流れを変えることができます (図 2)。

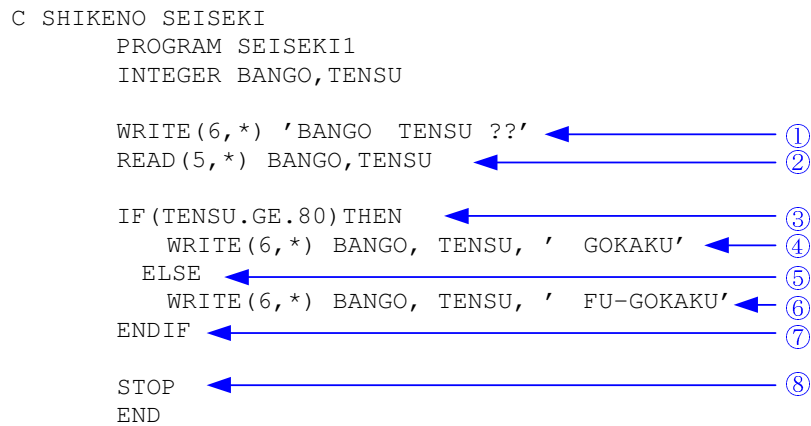


図 2 このプログラムの処理の順序は TENSU の値により、①②③④⑧の場合と①②③⑥⑧の場合がある。

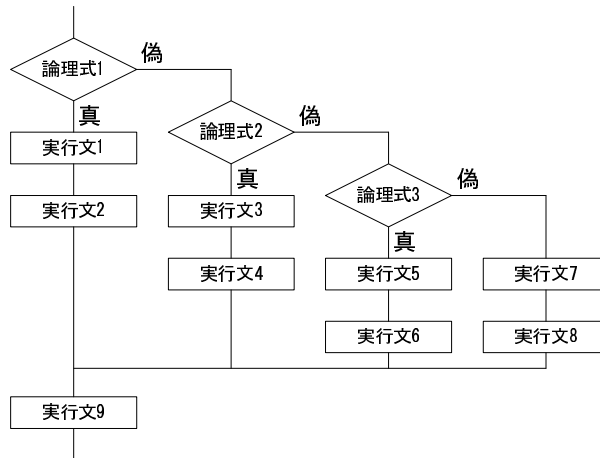
- コンピューターはアホです。人間が書いたプログラムに従い 1 行ずつ順番に、その内容を実行しているだけです。単純な命令を非常に高速に処理してます。この世の中にあるいかなるプログラムの同じです。

2 IF 文

- IF 文は、論理式の値によりプログラムの流れを変える。論理式の値によりプログラムの流れが変わることをよく理解すること。
- IF 文は、①ブロック IF 文②論理 IF 文③算術 IF 文がある。これらの違いを理解すること。
 - ブロック IF 文 (ELSE IF や ELSE が無い場合もある)

```

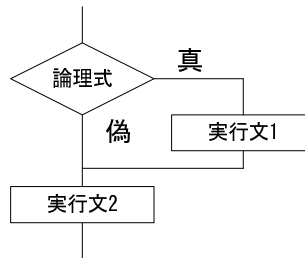
IF (論理式 1) THEN
    実行文 1
    実行文 2
ELSE IF (論理式 2) THEN
    実行文 3
    実行文 4
ELSE IF (論理式 3) THEN
    実行文 5
    実行文 6
ELSE
    実行文 7
    実行文 8
ENDIF
    実行文 9
    
```



- 論理 IF 文

```

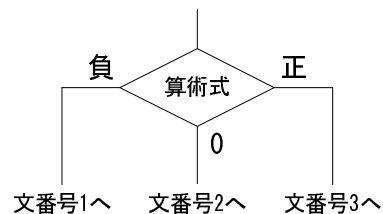
IF (論理式) 実行文 1
    実行文 2
    
```



- 算術 IF 文

```

IF (算術式) 文番号 1, 文番号 2, 文番号 3
    
```



- ブロック IF 文には、以下のパターンがある。IF 文のプリントに書いてあるので、違いを理解しておくこと。

- IF (論理式) THEN～ENDIF
- IF (論理式) THEN～ELSE～ENDIF
- IF (論理式) THEN～ELSE IF (論理式) THEN～ELSE IF (論理式) THEN～ELSE～ENDIF

3 論理式を構成する関係演算子

- 算術式と論理式の違いを理解すること。算術式の結果は数値、それに対して論理式の結果は論理値の真 (True) または偽 (False) のいずれかである。
- 以下の関係演算子は絶対に憶えること。IF 文の論理式に使われる。

関係演算子。A と B との比較

数学	FORTRAN	演算子	語源
$A < B$	A.LT.B	.LT.	Less Than
$A \leq B$	A.LE.B	.LE.	Less than or Equal
$A = B$	A.EQ.B	.EQ.	Equal
$A \neq B$	A.NE.B	.NE.	Not Equal to
$A > B$	A.GT.B	.GT.	Greater Than
$A \geq B$	A.GE.B	.GE.	Greater than or Equal

3 GO TO 文

- GO TO 1 のように使います。この実行文があると強制的に、文番号 1 の行に実行が移ります。
- 文番号とは、実行文の目印です。実行文の第 1 桁目から第 5 桁目まで、任意の 1 個の整数を書くことができます。

4 ループ構造

- 同じ実行文を複数回繰り返す構造です。IF 文と GO TO 文を組み合わせるとこの構造を作ることができます。プリントを見てよく理解しておくこと。

5 初期化とカウンター

- 変数は宣言文によって、コンピューターのメモリー (主記憶装置) の一部が予約され使用可能になります。
- 使用可能になった状態では、その変数の値は未定です。どんな値になっているかわかりません。
- そこで、算術代入文を使って初期化します。例えば、変数 A をゼロにする場合は、以下の通り。

A=0

- 数を数える処理をカウンターと言います。ひとつずつ数を数える場合は、

N=N+1

とします。これにより、A の値がひとつずつ増加します。もちろん、N は宣言文で整数と宣言をして、初期が必要です。

6 足し合わせ

- 同じような数字を何回も加算することがあります。例えば、

$$\text{SUM} = 1^2+2^2+3^2+4^2+5^2+6^2+7^2+8^2+9^2+\dots+100^2$$

などです。この場合は、100 回ループを回して、足し合わせます。足し合わせの部分の実行文は、

```
SUM=0
I=0
1  I=I+1
   SUM=SUM+I**2

   IF (I.LT.100) THEN
     GO TO 1
   END IF
```

のように書きます。初期化(SUM=0, I=0)して、ループ構造を用いて、足し合わせる(SUM=SUM+N**2)のです。

7 教科書の練習問題(P.48~P.49)

- 2-A (1) 最重要
- 2-A (2) 整数同士の演算は、たとえ除算でも整数になること忘れないで、プログラムの流れを追ってみる。

		L の値	K の値
	L=9	9	?
	L=(L/4)*4	①9/4 は整数の演算なので、9/4→2 ②次に 2*4→8	8 ?
	L=L-1	7	?
5	K=L-2*L/5	①7-2*7/5 の演算 ②7-(2*7)/5 の順序で計算する。 ③7-14/5 ④7-2→5	7 5
	IF (K) 1, 5, 4	①K は正なので文番号 4 を実行	7 5
4	L=L-1	①7-1→6	6 5
	K=K-1	②5-1→4	6 4
1	STOP	6	4
	END		

- 2-B (1) ブロック IF 文、論理 IF 文、算術 IF 文の使い方を理解すること。
- 2-B (2) 学習の範囲を超えている部分があるので試験範囲外とします。
- 2-B (3) 学習の範囲を超えている部分があるので試験範囲外とします。

8 実際のプログラム

- 今まで学習した内容が全て含まれているプログラムです。このプログラムが書けるようになってください。
- 1～N までのべき乗の和を求めるプログラムを作成する。キーボードから、和を求める最大の N の値を読み込んで、

$$\begin{aligned}S1 &= 1+2+3+4+5+6 \cdots N \\S2 &= 1^2+2^2+3^2+4^2+5^2+6^2 \cdots N^2 \\S3 &= 1^3+2^3+3^3+4^3+5^3+6^3 \cdots N^3\end{aligned}$$

を計算する。N と計算結果である S1, S2, S3 をディスプレイに書き出す。ディスプレイに書き出したならば、再度 N を読み込み同様の計算と表示を行う。もし、N ≤ 0 が入力されたならば、プログラムの動作を終了させる。

```
C  WA NO KEISAN
    PROGRAM SUN
    INTEGER I, N
    INTEGER S1, S2, S3

C  --- SHOKIKA ---
1  I=0
    S1=0
    S2=0
    S3=0

    WRITE(6,*) 'N= ??'
    READ(5,*) N

    IF(N.LE.0) THEN
        STOP
    ENDIF

2  I=I+1

    S1=S1+I
    S2=S2+I*I
    S3=S3+I**3

    IF(I.LT.N) THEN
        GO TO 2
    ELSE
        WRITE(6,*) N, 'MADE NO WA'
        WRITE(6,*) 'SUM I   =', S1
        WRITE(6,*) 'SUM I**2 =', S2
        WRITE(6,*) 'SUM I**3 =', S3
        GO TO 1
    ENDIF

END
```