

後期中間試験について

山本昌志*

2003年11月25日

1 試験について

1.1 出題範囲

試験範囲は、教科書の3章です。3章の主な内容は、以下の通りです。これを十分理解して、試験に臨むこと。

- DO文によるループ
- 1次元配列 (DIMENSION)
- 書式付出力 (FORMAT文)

1.2 出題内容

以下のような問題を出題する。

1. 1～Nまでの和の計算。IFとGOTO文を使ったプログラムと、DO文を使ったもの虫食い問題(2)章
2. DO文の文法
3. DO文の使い方
4. 一次元配列の宣言の仕方を問う問題
5. 一次元配列へのデータの入れ方を問う問題
6. 一次元配列からのデータの取り出し方を問う問題
7. FORMAT文の使い方を問う問題。ただし、WRITE文に限ります。
8. 1～Nまでの奇数の和。このプリントの最後のプログラムを記述する問題をを出します。
9. 教科書の練習問題 (p.78～p.79)

*国立秋田工業高等専門学校 電気工学科

- 3-A(1)
- 3-A(2)
- 3-A(4)
- 3-A(7)
- 3-A(8)

以下については、教科書に記述はありますが、現在ではほとんど使われませんので、試験には出しません。

- ラインプリンターの制御
- カードからのデータ読み込み

2 DO文

コンピュータープログラム上で同じ処理を繰り返す、ただし変数などの値は異なっているようなものをループといいます。これは、以前学習した IF 文と GO TO 文で実現できたことを思い出してください。例えば、ループ構造を使って $1 \sim N$ までの和を計算するプログラムは、次のようになります。

```

PROGRAM SUM N BY USING IF AND GO TO
INTEGER N,I,S

I=0
S=0

READ(5,*)N
1  I=I+1
   S=S+I
   IF(I.LT.N)THEN
     GO TO 1
   ELSE
     WRITE(6,*)S
     STOP
   ENDIFIF

END

```

10

コンピューターは同じようなことを繰り返すのが非常に得意です。このループ構造は、いたるところで出現します。FORTRAN ではループを構造を作る専用の実行文があります。それが、ここで学習する DO 文というものです。先の IF 文と GO TO 文で作成した同じプログラムを DO 文で書くと、以下のようになります。先に比べて、プログラムは非常にすっきりしたものになります。

```

PROGRAM SUM N BY USING DO
INTEGER N,I,S

I=0
S=0

```

```

READ(5,*)N

DO 10 I=1,N,1
  S=S+I
10 CONTINUE

WRITE(6,*)S
STOP

END

```

10

それでは、新しい実行文 DO と CONTINUE について、説明します。まずは、先ほどのプログラムのループ構造の説明の図 1 を見てください。DO と CONTINUE は対になって表れ、その間がループになっていることが分かります。単純でしょう。

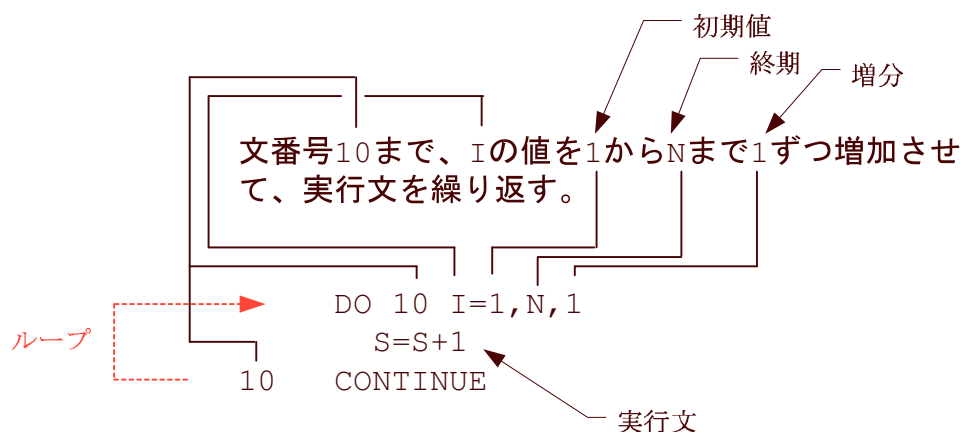


図 1: DO 文を使ったループ構造

教科書の P.57~60 もよく学習しましょう。

3 1次元配列

3.1 配列とは

コンピューターでデータを処理するときには、データは全て変数に入れて処理します。変数とはデータを入れる箱のことです。その箱の中には、型 (REAL や INTEGER など) が指定されたデータがたった一つ入ります。

ここで問題が生じます。整数のデータが 1 万個あった場合、どうしますか?。一つの解決方法は、1 万個の整数の型の変数を用意することです。例えば、

```

INTEGER AAA, AAB, AAC, AAD, AAE, AAF, AAG, AAH
INTEGER AAI, AAJ, AAK, AAL, AAM, AAN, AAO, AAP

```

C まだまだ変数宣言をいっぱい書く

C 変数ごとの処理を書く

END

のように書きます。この方法は明らかに手間が必要で、現実的では有りません。問題の所在は、

- 10000 個の変数を 1 個ずつ宣言している。
- その 10000 個の変数の、個々について処理をしようとしている。

というところにあります。明らかにプログラムが大変です。

こういうときに配列と呼ばれる変数の入れ物を使います。それは、

INTEGER ABC(10000)

のように宣言します。すると、ABC(1), ABC(2), ABC(3), ..., ABC(9999), ABC(10000) のような、整数の入れ物が用意されます。そして、その入れ物の中のデータ (個々では整数の数値) を操作したい場合は、ABC(50) というようにすればよいのです。配列を使うと、大きなデータを扱う場合、非常に簡単になります。ここで、ここで言葉の定義をしっかりとっておきましょう。先ほどの例を使って、

- ABC を配列名といいます。
- 10000 のように配列に格納できるデータ数を配列の大きさと言います。
- ABC(1) や ABC(2)、ABC(7777) などの個々の配列のデータを配列の要素と言います。ここでは、配列の要素は 10000 個有ります。
- 要素を示す番号を添字と言います。

と、言葉は定義されます。

3.2 配列の宣言の方法

配列を使うときには宣言が必要です。以前学習した変数の宣言文と同様に、配列の宣言文を書かなくてはなりません。宣言文なので、実行文に先立って書く必要があります。例えば、

配列の型 配列名 配列の大きさ (下限:上限)

となります。配列の下限を省略した場合、それは 1 となります。実際には次のような宣言の方法があります。通常は、以下の宣言の方法を用います。

REAL A(1:3) 実数型の配列、A(1),A(2),A(3) を用意する。
REAL B(-1:2) 実数型の配列、B(-1),B(0),B(1),B(2) を用意する。
REAL C(5:6) 実数型の配列、C(5),C(6) を用意する。
REAL D(3) これは、配列の下限を省略した場合で、それは 1 となります。従って用意される配列は、実数型の D(1), D(2), D(3) です。

宣言として、DIMENSION を使う方法もあります。その場合、変数宣言ができませんので暗黙の型宣言¹に従います。あるいは、DIMENSION のあとに配列名の型宣言を行えば、型を決められます。ただし、先ほどの方法に比べて 2 行書く必要があり、手間がかかります。例えば、以下のように記述すると暗黙の型宣言が適用されます。

DIMENSION A(1:3)	実数型の配列、A(1),A(2),A(3) を用意する。
DIMENSION KVALUE(-1:2)	整数型の配列、KVALUE(-1),KVALUE(0),KVALUE(1),KVALUE(2) を用意する。
DIMENSION RVALUE(3)	実数型の配列 RVALUE(1),RVALUE(2),RVALUE(3) を用意する。

この暗黙の型宣言を避けるためには、以下のように記述します。

DIMENSION KVALUE(-1:2)	実数型の配列、KVALUE(-1), KVALUE(0),
REAL KVALUE	KVALUE(1), KVALUE(2) を用意する。
DIMENSION RVALUE(3)	整数型の配列 RVALUE(1), VALUE(2), RVALUE(3)
INTEGER RVALUE	を用意する。

3.3 配列の使い方

教科書の P.60~68 でよく読んでください。

4 書式付出力 (FORMAT)

4.1 WRITE と FORMAT

FORTRAN のプログラムの出力には、WRITE 文が使われる。書式を指定して、出力をしたい場合、FORMAT 文を使う。例えば、

```
WRITE(6,600)2*M-1,S
600  FORMAT('1 KARA ',I5,' MADE NO KISU WA S=',I9)
```

のように書きます。WRITE(6,600) の 6 が出力先 (ディスプレイ)、600 が書式を書いてある文番号を示します。

¹配列名の先頭が I,J,K,L,M,N で始まる場合、その配列の型は INTEGER(整数) になる。それ以外の場合、配列の型は REAL(実数) になる。

4.2 編集記述子

以下の編集記述子の使い方を理解すること (教科書の p.72 以降)。

表 1: よく使われる編集記述子。小文字は整数を表す。

記号	説明
nH	H の次に続く文字 n 個を出力する
'...'	アポストロフィで囲まれた文字を出力する
nX	n 個の空白を入出力
Fw.d	実数の入出力 (小数) w:全体の桁数 d:小数部の桁数
Ew.d	実数の入出力 (E 型) w:全体の桁数 d:小数部の桁数
Iw	整数の入出力 w:全体の桁数
Iw.m	整数の入出力 w:全体の桁数 m:少なくとも m 桁出力

5 プログラム

以下の奇数の和を計算するプログラムを作成しなさい。

$$S = 1 + 3 + 5 + \dots + N$$

作成するプログラムは、以下のような機能を有すること。

- 先の奇数の和を計算すること。
- N はキーボードから読み込む。もし N が偶数ならば、N-1 までの和とする。
- 加算すべき数字は、一次元配列に格納する。一次元配列の要素数は 10000 とする。
- DO 文を使って、一次元配列の値を読み出して、加算すること。
- 結果は、以下のように出力すること。

```
1 KARA 5 桁整数 MADE NO KISU WA S= 9 桁整数
```

最初の 5 桁の整数は加算する最大の奇数、次の 9 桁の整数は和を表す。

これを実現するプログラムは、例えば次のようになります。

```
PROGRAM EXERCISE
```

```

INTEGER N,S,I
INTEGER A(1:10000)

READ(5,*)N

M=(N+1)/2

DO 10 I=1,M,1
10  A(I)=2*I-1
   CONTINUE

S=0

DO 20 I=1,M,1
   S=S+A(I)
20  CONTINUE

WRITE(6,600)2*M-1,S
600 FORMAT('1 KARA ',I5,' MADE NO KISU WA S=',I9)

STOP
END

```