

情報処理概論(第7回)

情報基盤センター 井上 仁

今日の授業内容(第7回)

- ベクトル・行列の計算
- 第一回レポートについて
- 変数の型(実数型、文字型)

変数の型(実数型1)

■ REAL, REAL(4) : (単精度)実数型

- ◆ 変数の宣言
 - REAL :: radius, area
- ◆ 4バイトの記憶領域
- ◆ 実数を近似的に表現
 - 2.0 3.141592 0.2785E2 → 0.2785×10^2
 - 2と書くと整数になる(内部表現も異なる)
- ◆ 有効桁数
 - 10進で約6桁(2進で24桁)
- ◆ 絶対値
 - 約 $10^{(-37)}$ ~ 約 10^{38} の値($2^{(-126)}$ ~ 2^{127})

変数の型(実数型2)

■ REAL(8) : 倍精度実数型

- ◆ 8バイトの記憶領域(倍精度)
 - ◆ 実数を近似的に表現
 - 3.1415926535 0.2785D-2 → $0.2785 \times 10^{(-2)}$
 - ◆ 有効桁数
 - 10進で約14桁(2進で53桁)
 - ◆ 絶対値
 - 約 $10^{(-307)}$ ~ 約 10^{308} の値($2^{(-1022)}$ ~ 2^{1023})
- ※REAL(16) 4倍精度実数型

例題(ベクトルの和)

■ 二つのn次元ベクトルa, bの和を求める

$a = (a_1, a_2, \dots, a_n)$
 $b = (b_1, b_2, \dots, b_n)$
 $sum = (a_1+b_1, a_2+b_2, \dots, a_n+b_n)$
 ⇒ 各 $i(i=1, \dots, n)$ に対して、 a_i+b_i を求める



```

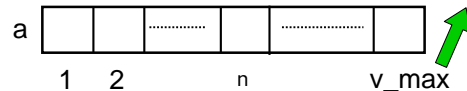
DO i = 1, n
  sum(i) = a(i) + b(i)
END DO
    
```

変数の宣言

~~INTEGER :: v_max = 20~~
 初期値
 INTEGER, PARAMETER :: v_max = 20
 名前付き定数

REAL :: a(1:v_max), b(1:v_max)
 または
 REAL, DIMENSION(1:v_max) :: a, b

このプログラムで扱うことのできるベクトルの次元



プログラムの流れ

- ベクトルの次元の入力と入力値のチェック
 - ◆プログラムではv_max (20) 次元まで扱うことができる
 - ◆実際に取り扱うベクトルの次元nを入力する
- 二つのベクトルの各成分の入力
- 二つのベクトルの和の計算
- 結果(和)の出力

入力値のチェック

```

■ベクトルの次元は、1以上v_max以下
DO
  READ(*,*) n
  IF (1 <= n .AND. n <= v_max) EXIT
END DO
1 ≤ n ≤ v_max
    
```

DO形並びによる入力

```

DO i=1, n
  READ(*,*) a(i)
END DO

```

1
2
3
 といった入力しかできない

~~READ(*,*) a(1), a(2), ..., a(n)~~

READ(*,*) (a(i), i=1, n)

1 2 3 という入力が可能

DO形並びによる出力

```

DO i=1, n
  WRITE(*,*) sum(i)
END DO

```

1
2
3
 といった出力しかできない

~~WRITE(*,*) sum(1), sum(2), ..., sum(n)~~

WRITE(*,*) (sum(i), i=1, n)

1 2 3 という出力が可能

プログラム例 (ベクトルの内積)

内積 $s = a_1*b_1 + a_2*b_2 + \dots + a_n*b_n$

$s = 0$

```

DO i = 1, n
  s = s + a(i)*b(i)
END DO

```

$s = s + a(n)*b(n)$

例題(行列の和)

- m行n列の二つの行列の和を計算
- A : m行n列の行列 a(1:m,1:n) : 2次元の配列

	1	2	3	n-1	n	
a(1,1)	a(1,2)			a(1,n)	1	
a(2,1)					2	
⋮	⋮	⋮	a_{ij}	⋮	⋮	⋮	
a(m,1)				a(m,n)	m	

解法の説明

■ A, Bを二つの行列、Cをその和とする。

$$C = A + B$$

$$c_{ij} = a_{ij} + b_{ij} \quad (i = 1, \dots, m \quad j = 1, \dots, n)$$

◆ i=1に対して

$$c_{1j} = a_{1j} + b_{1j} \quad (j = 1, \dots, n)$$

◆ i=2に対して

$$c_{2j} = a_{2j} + b_{2j} \quad (j = 1, \dots, n)$$

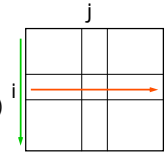
◆

◆ i=mに対して

$$c_{mj} = a_{mj} + b_{mj} \quad (j = 1, \dots, n)$$

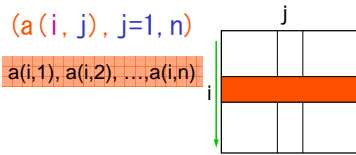
プログラム例

```
DO i=1, m (row)
  DO j=1, n (column)
    c(i, j) = a(i, j) + b(i, j)
  END DO
END DO
```



DO形並び

```
DO i=1, m
  READ (*, *) (a(i, j), j=1, n)
END DO
```



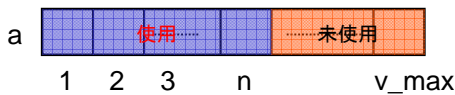
```
DO i=1, m
  WRITE (*, *) (c(i, j), j=1, n)
END DO
```

c(i,1), c(i,2), ..., c(i,n)

実は...

■ 配列 a, b に対して、a+b という計算ができる

配列の動的割り付け



■ 動的割り付け

型, ALLOCATABLE :: 配列変数名 (:)

.....

ALLOCATE (配列変数(寸法指定))

■ 割り付け解除

DEALLOCATE (配列変数)

プログラム例

```
REAL, DIMENSION (:), ALLOCATABLE :: a, b
.....
READ (*, *) n
ALLOCATE (a(1:n))
ALLOCATE (b(1:n))
.....
DEALLOCATE (a)
DEALLOCATE (b)
```

Tips

- 画面の入出力のファイルへの保存
 - ◆ script または script ファイル名
 - % script
 - script コマンド開始。ファイルは typescript です。
 - % ls
 - a.out* do2.f90 ex1_1b.f90
 -
 - % exit
 - script コマンド終了。ファイルは typescript です。
 - %

演習7-1

- ベクトルの和を計算するプログラムを完成させなさい
- ベクトルの内積を計算するプログラムを完成させなさい
- ベクトルを入力して、それを正規化したベクトルを出力するプログラムを作成しなさい。

演習7-2

- 行列の和を計算するプログラムを完成させなさい
- $m \times n$, $n \times l$ の行列を入力して、その積を計算するプログラムを作成しなさい。

演習7-3

- 演習7-1, 7-2のプログラムにおいて、配列を動的に割り付けるよう変更しなさい。

レポート課題

- 課題
 - ◆ 非負整数を入力し、それを16進数に変換したものを出力するプログラムを作成しなさい。
- 提出物
 - ◆ Fortran90のソースプログラム、プログラムの説明、実行結果、考察
 - ※これらのファイルは、別のファイルにしてもよいし、一つのファイルにしてもよい。
- 提出方法
 - ◆ WebCTのレポート機能を利用する
- 提出期限
 - ◆ 6月24日(金)

10進数から16進数への変換

60 (10進数) → 3C (16進数)
 2002 (10進数) → 7D2 (16進数)

$\begin{array}{r} 16 \overline{) 60} \\ \underline{48} \\ 12 \\ \underline{12} \\ 0 \end{array}$	$\cdots 12 \uparrow$ $\cdots 3 \uparrow$	$\begin{array}{r} 16 \overline{) 2002} \\ \underline{1280} \\ 722 \\ \underline{640} \\ 82 \\ \underline{64} \\ 18 \\ \underline{16} \\ 2 \\ \underline{0} \\ 0 \end{array}$	$\cdots 2 \uparrow$ $\cdots 13 \uparrow$ $\cdots 7 \uparrow$
--	---	---	--

変数の型 文字型(1)

■ CHARACTER

- ◆ 0~32767個の文字の列
- ◆ 1バイトの記憶域に1文字
 - 日本語文字の場合は、2バイトで1文字
 - 文字コードによって、前後に制御シーケンス

■ CHARACTER(n)

- ◆ nバイトの文字列
 - 'Fortran' 'FORTRAN' 'Don' 'NT' "Don' t"
 - '日本語'

変数の型 文字型(2)

CHARACTER(10) :: a

a = 'Fortran'

F	o	r	t	r	a	n			
---	---	---	---	---	---	---	--	--	--

WRITE(*,*) a, a

% ./a.out

Fortran Fortran

漢字コード

■ 'Fortran'

- ◆ 46 6f 72 74 72 61 6e (ASCII)

■ '日本語'

- ◆ 46 7c 4b 5c 38 6c (JIS)
- ◆ 符号化方式: JIS, Shift JIS, EUC
 - 1b 24 42 46 7c 4b 5c 38 6c 1b 28 42 : JIS
 - 93 fa 96 7b 8c ea: Shift JIS
 - c6 fc cb dc b8 ec : EUC

■ EUC を使う