

Fortran Week 3

- HW11解答：
`/home/teachers/fortran_ta/data/PSC2020/hw11/`
- 之前所有上課練習、實作課的答案：
`/home/teachers/weitingc/work/`
- 本週影片問題
- hw12作業說明

本週預習作業 (1)

- 請問 $b(2:3) = ?$

```
PROGRAM PREVIEW3
```

```
IMPLICIT NONE
```

```
INTEGER, PARAMETER :: MAX = 10
```

```
INTEGER :: i
```

```
REAL, DIMENSION(MAX) :: a = (/ (REAL(i*2), i=1,10) /)
```

```
REAL, DIMENSION(MAX/2) :: b
```

```
b = exp(a(3:7))
```

```
END PROGRAM PREVIEW3
```

• $a = /2, 4, 6, 8, \dots, 20/ \rightarrow a(3:7) = /6, 8, 10, 12, 14/$

$\rightarrow b(1:5) = /exp(6), exp(8), exp(10), exp(12), exp(14)/$

$\rightarrow b(2:3) = /exp(8), exp(10)/$

本週預習作業 (2)

- (pr4.f95) 將pr4_in.txt 讀進一個 6×8 陣列中，將左下角四分之一轉置寫入pr4a_out.txt，要用到迴圈，輸出格式 $\square\square0.14E+01\square\square0.16E+01\square\square0.18E+01$
- (pr4_in.txt) $A(1:6, 1:8) \rightarrow$ 左下四分之一 = $A(4:6, 1:4)$

0.000, 0.693, 1.098, 1.385, 1.607, 1.788, 1.940, 2.071
0.693, 1.386, 1.791, 2.077, 2.299, 2.479, 2.631, 2.762
1.099, 1.791, 2.196, 2.482, 2.704, 2.884, 3.035, 3.166
1.386, 2.079, 2.484, 2.770, 2.991, 3.171, 3.322, 3.452
1.609, 2.302, 2.707, 2.993, 3.214, 3.394, 3.545, 3.674
1.792, 2.484, 2.889, 3.175, 3.396, 3.576, 3.726, 3.856

- 讀:

```
DO i=1,6
  READ(10,*) (A(i,j), j=1,8)
ENDDO
```

- 寫:

```
DO j=1,4
  WRITE(20,10010) (A(i,j), i=4,6)
ENDDO
10010 FORMAT(3(2x,E8.2e2))
```

提問-Do While

```
REAL :: k=1., S=0
DO WHILE (k>0)
    WRITE(*,*) 'Please input next number (>0):'
    READ(*,*) k

    IF (k>0) THEN
        S=S+k
        WRITE(*,*) 'Sum =' , S
    ENDIF

END DO
```

Q：在上方DO WHILE迴圈中不是偵測到輸入的K值不大於0就會自動跳出了嗎？為什麼還需要一個IF迴圈？

A：接收輸入的K值是在進入DO WHILE迴圈之後，無論輸入K值多少，還是會執行完整段迴圈，返回DO WHILE (K>0)這行才根據K值決定要繼續或終止迴圈

補充說明-1 Do 迴圈—Cycle功能

- 直接跳回迴圈的開頭，執行下一次迴圈。舉例：

```
PROGRAM test_cycle
IMPLICIT NONE
INTEGER :: I

DO I=1,5
    IF (I==3) THEN
        CYCLE ! Go back to top and continue
    ENDIF
    WRITE(*,*) I
ENDDO

WRITE(*,*) 'End Loop!'

END PROGRAM test_cycle
```

當迴圈進行到counter $I=3$ 的時候，不執行之後的WRITE指令，直接跳回迴圈開頭，並且令 $I+1 (=4)$ 繼續迴圈，直到迴圈結束

DO i=1,5

IF (i==3)

(i \neq 3)

CYCLE

(i=3)

ENDIF

WRITE(*,*) i

ENDDO

(i+1)

(i+1)

補充說明-2 格式碼

- **format**中，整數變數的寫法是**Iw.d**，**d**的用途是什麼？是否可以省略
- **d**是"一定要顯示的位數有幾位"，如果不足的話會**補零**，直接看例子比較容易懂
- 用 **I5.3**來顯示整數1→ **□□001**
「總寬度5個字元，至少要顯示3個位數」，所以會補兩個0
- 用 **I5.2**來顯示整數1→ **□□□01**
「總寬度是5個字元，至少要顯示2個位數」
- 用 **I5**來顯示整數1→ **□□□□1**
「總寬度是5個字元」，不補0

補充說明-3 利用格式碼將數值轉為字串

舉例：想要開啟檔案202001.txt, 202002.txt, 202003.txt...202012.txt

- 步驟一：建立迴圈，讓counter從202001增加到202012
- 步驟二：利用**WRITE**與**格式碼**，將**counter**的數值”寫入“對應的字串變數
- 步驟三：利用轉換好的字串組合出檔名字串、開啟檔案

INTEGER :: **i**

CHARACTER(8) :: **str_i**

CHARACTER(12):: **fname**

Do **i** = 202001, 202012

WRITE(str_i, 'i6') **i** ! converting integer i to string str_i

fname = trim(**str_i**)//'.txt'

OPEN(10, fname, status='unknown')

...

ENDDO

補充說明-4 隱式迴圈

- 2D Array 第9頁中，要將一個文字檔分別讀取成兩個陣列。

文字檔 (data2.txt)

A1	B1	A2	B2	A3	B3
1.1,	1.2,	1.3,	1.4,	1.5,	1.6
2.1,	2.2,	2.3,	2.4,	2.5,	2.6

```
PROGRAM 2DArray_read2
REAL, DIMENSION(2,3) :: A=0., B=0.
INTEGER :: I, J
OPEN(10,file='array2.txt')
READ(10,*) ! pass header in file
DO I=1,2
    READ(10,*) (A(I,J),B(I,J), J=1,3)
END DO
WRITE(*,*) A(2,3), B(1,2)
END PROGRAM 2DArray_read2
```

如何將資料分別讀入
A與B兩個2x3的陣列？

A(1,1) =1.1	A(1,2) =1.3	A(1,3) =1.5
A(2,1) =2.1	A(2,2) =2.3	A(2,3) =2.5

B(1,1) =1.2	B(1,2) =1.4	B(1,3) =1.6
B(2,1) =2.2	B(2,2) =2.4	B(2,3) =2.6

補充說明-4 隱式迴圈

- 2D Array 第9頁中，要將一個文字檔分別讀取成兩個陣列。

文字檔 (data2.txt)

A1	B1	A2	B2	A3	B3
----	----	----	----	----	----

1.1, 1.2, 1.3, 1.4, 1.5, 1.6 → A(1,1), B(1,1), A(1,2), B(1,2), A(1,3), B(1,3)

2.1, 2.2, 2.3, 2.4, 2.5, 2.6 → A(2,1), B(2,1), A(2,2), B(2,2), A(2,3), B(2,3)

```
PROGRAM 2DArray_read2
```

```
REAL, DIMENSION(2,3) :: A=0., B=0.
```

```
INTEGER :: I, J
```

```
OPEN(10,file='array2.txt')
```

```
READ(10,*) ! pass header in file
```

```
DO I=1,2
```

```
    READ(10,*) (A(I,J), B(I,J), J=1,3)
```

```
END DO
```

```
WRITE(*,*) A(2,3), B(1,2)
```

```
END PROGRAM 2DArray_read2
```

A(1,1) =1.1	A(1,2) =1.3	A(1,3) =1.5
A(2,1) =2.1	A(2,2) =2.3	A(2,3) =2.5

B(1,1)= 1.2	B(1,2)= 1.4	B(1,3)= 1.6
B(2,1)= 2.2	B(2,2)= 2.4	B(2,3)= 2.6

```
DO I=1 ... READ (J=1~3)
```

A(1,1), B(1,1), A(1,2), B(1,2), A(1,3), B(1,3)

```
DO I=2 ... READ (J=1~3)
```

A(2,1), B(2,1), A(2,2), B(2,2), A(2,3), B(2,3)

補充說明-5 MAXLOC函數

- MAXLOC回傳的是一維整數陣列
- 可以直接做為subscript使用，取出陣列的元素
 - **WRITE(*,*) x(MAXLOC(y))**
- 如果用MAXLOC回傳的結果對單一變數(非陣列)賦值，會造成dimension不合的問題
 - **INTEGER :: n**
...
n= MAXLOC(y)
- 正確的宣告方法
 - **INTEGER, DIMENSION(1) :: n**
...
n= MAXLOC(y)

編譯時出現錯誤訊息

n=MAXLOC(y)

1

Error: Incompatible ranks 0 and 1
in assignment at (1)

補充說明-6 陣列外積(outer product)

- 注意：FORTRAN中兩個陣列相乘，結果會是個別”元素對元素”的相乘，與數學上向量的內積或外積的定義均不相同

```
REAL, DIMENSION(3) :: a=(/1.,2.,3./),  
b=(/5.,6.,7./)
```

```
REAL, DIMENSION(3) :: c
```

```
c = a * b
```

$$c(1) = a(1) * b(1) = 5.$$

$$c(2) = a(2) * b(2) = 12.$$

$$c(3) = a(3) * b(3) = 21.$$

- 算內積：用內建函數 dot_product (a, b長度要相同)

```
d = dot_product(a, b)
```

$$d = a(1)*b(1)+a(2)*b(2)+a(3)*b(3)$$

- 算外積：無內建函數，要自己寫**subroutine/function**
(範例: /home/teachers/weitingc/lecture_ex/out_product.f95)

- Matrix Outer Product
- Given \mathbf{u} is a $m \times 1$ column vector and \mathbf{v} is a $n \times 1$ column vector
- Their outer product is defined as the $m \times n$ matrix \mathbf{A} , which is obtained by multiplying each element of \mathbf{u} by each element of \mathbf{v} :

$$\mathbf{u} = (u_1, u_2, \dots, u_m)$$

$$\mathbf{v} = (v_1, v_2, \dots, v_n)$$

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \\ u_4 v_1 & u_4 v_2 & u_4 v_3 \end{bmatrix}$$

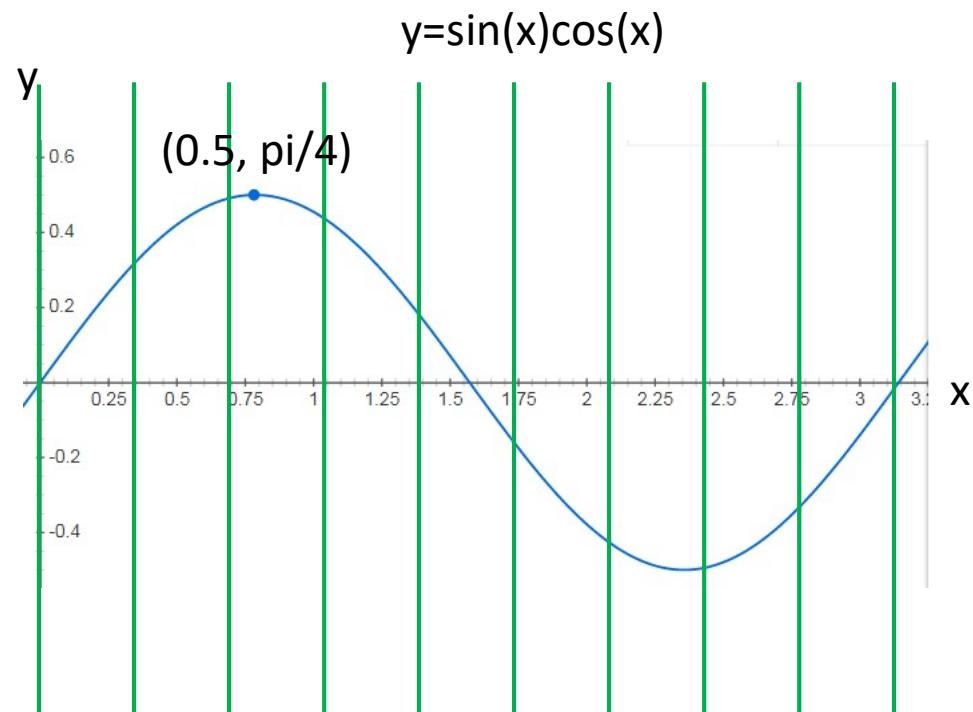
示範: 陣列內建函數

- /home/teachers/weitingc/work/work6.f95

```
INTEGER, PARAMETER :: NX=10
REAL, DIMENSION(NX) :: x, y
REAL, PARAMETER :: pi=3.1415926
INTEGER :: i

DO i=1,NX
    x(i)=pi*(i-1.) / (NX-1.)
ENDDO
y=sin(x)*cos(x)
```

增加為100, 1000, 10000
看結果有何不同



- 加上程式碼，在螢幕上顯示
y的最大值
y的最大值發生在第幾個元素
y的最大值所對應的x值