

C程式語言

後測式重複結構 do-while

巢狀重複結構 nested loop

陣列 array


陳佳宜

Review

while loop
do-while loop

前測式重複結構：執行前先檢查條件

每次迴圈執行前要檢查的條件



```
while ( some condition ){  
    //要執行的指令內容寫在這兒;  
}
```

後測式重複結構：執行後才檢查條件

- 先執行迴圈內指令後才判斷
- 迴圈至少會執行一次
- while(條件式)後要加分號

```
do{  
    ..../* Do something here */;  
} while ( some condition );
```



作業講解



SingContest (歌唱比賽)

題目說明

- 輸入
 - 五位評審的成績，最高為100分，僅採計至整數位
- 輸出
 - 評審的分數在輸出前須顯示目前為第幾位評審(第幾筆資料)
 - 輸出「總得分：」文字後，顯示得分



HealthyWeight (體位控制獎金挑戰賽)

題目說明

- 獎金發放金額
 - NT \$500元/次
- 獎金停止發放條件
 - 該次體重未比前次低
 - 已低於最低標準體重
 - BMI < 18.5

國民健康署建議BMI值：

過輕	BMI < 18.5
標準	18.5 ≤ BMI < 24
過重	BMI ≥ 24
打錯	BMI < 1, BMI > 50

巢狀重複結構 / 巢狀迴圈

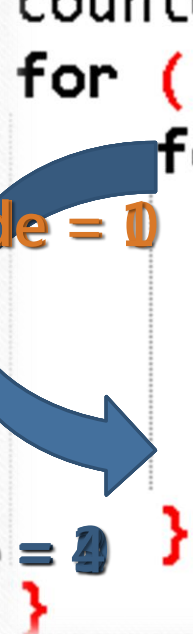
nested loop

迴圈裡包含另一個迴圈

可視任務需要任意搭配各種迴圈

→ for / while / do-while

觀察以下程式碼，巢狀迴圈是如何運作的？



```
int outside,inside,counter;
counter = 1;
for ( outside = 0; outside < 5; outside++ ){
    for ( inside = 0; inside < 5; inside++ ){
        printf( "\n*****第 %d 圈*****\n",counter);
        printf( "Outside = %d\n",outside); //外圈迴圈變數的值
        printf( "Inside = %d\n",inside); //內圈迴圈變數的值
        counter++;
    }
}
```

- 外層的 for 完成一次 (outside = 0)，內層的 for 將完整執行一輪 (inside = 0~4)



課堂演示

一次性完整列出九九乘法表

作業：直角三角形產生器 (1/2)

✎ 學藝股長準備設計班級布告欄，他計畫將布告欄的四個角落以螢光星星堆積成直角三角形的樣式。

購買教室布置的材料須事先請款，為了決定到底要貼幾層的螢光星星，請設計一個程式顯示層數不同的星星排列起來的樣子，星星以 * 符號表示。

輸入輸出範例 I

範例輸出

*

**

範例輸入

• 5

輸入輸出範例 II



```
Triangle.exe
10 輸入
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
```

陣列

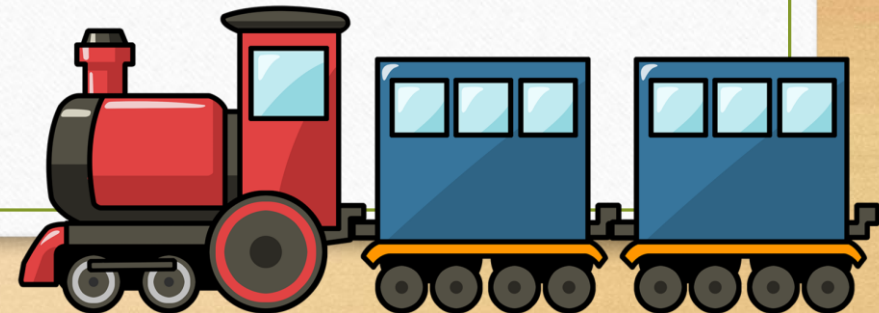
Array

陣列 (Array) 是什麼?

- 被賦予單一名稱的一連串相同資料型態
 - e.g 名稱為score的整數陣列

Index	0	1	2	3	4	5
Value	95	57	30	20	80	100

- 如何取得指定位置的值，陣列名稱[陣列編號]
 - e.g score[2]為30



如何定義一個整數陣列?

- 資料型態 陣列名稱[陣列元素的數量];
 - `int score[6];`
- 宣告陣列變數時，可同時給予初始值
 - `int score [6] = {95,57,30,20,80,100};` //指定陣列元素數量後，給定陣列元素內容
 - `int score[] = {95,57,30,20,80,100};` //給定陣列元素內容，系統依此給定陣列空間
- 使用{ }時，未初始化的部分會自動補成0
 - `int score [6] = {95,57};` //除了已被指定的索引值外，其餘皆為0

如何讀取輸入至整數陣列?

- 仍須使用&，因為陣列中的內容由多個元素組成，每個元素皆可視為一個整數變數。
 - e.g. `scanf("%d", &score[0]);` //將使用者鍵入的值存入index為0的score陣列中。
- 讀取數字陣列的方法皆相同，但在讀取浮點數陣列時，請記得scanf的格式指定字`%f`是不同的喔！

字串

- `char name[8]`
- C語言無法確切紀錄陣列的大小，因此需要知道何時到達陣列的尾端

Index	0	1	2	3	4	5	6	7
Value	J	e	s	s	i	c	a	\0

哨兵字元 (Sentinel Character)：字元陣列的尾端。
\0是值為0的ASCII字元，通常稱為null字元



如何定義一個字串/字元陣列?

- `char name[] = {'J','e','s','s','i','c','a'};`
- `char name[] = {"Jessica"};`
- `char name[] = "Jessica";`
- 若在定義時已初始化，當未給定陣列元素數量時，以初始化陣列元素數量給定陣列空間。

如何讀取輸入至字元陣列?

- `name` 變數代表該字串第一個字元的位址，因此不需要使用`&`取址。
 - 輸入的字元將依序存入陣列中。
 - e.g. `scanf("%s", name);`
- 讀取單一字元並存進字串陣列中的單一索引處
 - 仍須使用`&`，因為陣列中的元素由單一的字元組成，視為存入某個字元變數中。
 - e.g. `scanf("%c", &name[3]);`



課堂演示

查詢所輸入的文章中，指定字母出現的次數。

作業：資料傳輸 錯誤檢測 (2/2)



✎ 兩部電腦在傳輸資料的過程中，可能因為不正常的運作使得傳輸錯誤，所以會再加上奇同位元(parity bit)檢測，以檢驗傳輸上是否有誤，方法如下：

- 若傳輸資料所包含的 "1" 為奇數個，則parity bit為0；
- 若傳輸資料所包含的 "1" 為偶數個，則parity bit為1。

題目說明

- 輸入資料共8位數
 - 11101001 → parity bit為0
 - 10110100 → parity bit為1
- 請根據所輸入的傳輸資料判斷parity bit為何？

輸入輸出範例 I

範例輸出

- 1

範例輸入

- 00011000

輸入輸出範例 II

```
ParityBit.exe"
11001000 輸入
0
Process returned 0 (0x0) execution time : 15.706 s
Press any key to continue.
```


The End

See you next week !

陳佳宜