

程式設計第十章



真理大學資工系 洪麗玲
llhung@mail.au.edu.tw

摘要



期中考檢討

結構

結構練習

作業

題目:字串迴文



問題描述：

給予一個英文字母與數字參雜的字串，長度限制 在 256 個字母內。請撰寫一程式輸出此字串的迴文字串，將英文字母大寫改為小寫，小寫改為大寫，數字則不改變。

輸入說明：

輸入為一個長度在 256 個字母內的英文字母與數字構成的字串。

輸出說明：

經轉換後的迴文字串，最後必須有換行字元。

輸入範例	輸出範例
bCd12345	54321DcB

3



題目:學生成績排序與平均



•題目:學生成績排序與平均

問題描述：請撰寫一個程式，使用者會輸入學生人數(最多50人)，以及每位學生的成績 (0-99)，程式會將成績的排序加上全部的平均輸出。

輸入說明：使用者輸入學生人數 n ，以及每位學生的成績 (0~99)。

輸出說明：根據使用者輸入的成績，顯示排序的結果 (由高到低)。

輸入範例	輸出範例
10 50 89 25 2 26 12 52 65 10 79	89 79 65 52 50 26 25 12 10 2 平均 40.80

4



題目:加密與解密



問題描述：請設計一編碼與解碼的程式，輸入字元與數字說明要做的工作，字元**e**代表加密，字元**d**代表解密；而其後的數字代表每個字母的調整距離。之後在輸入一字串，系統會回報加密後或解密後的結果。**PS.請注意**，只有字母(大小寫)要進行加密解密的動作，數字或特殊字元都不需要更改。

輸入說明：使用者輸入需求 **e**(encode加密) 或 **d**(decode解密) 與調整的距離後，再輸入要進行加密或解密的句子。

輸出說明：根據使用者的輸入的成績，顯示經過加密或解密的結果。

輸入範例	輸出範例
e 3 this is a test	wklv lv d whvw
d 3 wklv lv d whvw	this is a test

5



題目:列出每個字元出現的次數



問題描述：請建立函式search(int a, char *str, char *T)，用來查詢字串中所包含的每個字元各出現幾次。之後實際輸入句子執行查詢動作。

輸入說明：輸入有一長度在 50 個字元內的句子。

輸出說明：輸出該字串中每個字元各出現幾次(空白不算)

輸入範例	輸出範例
a quick brown fox jump over the fence	a 1, q 1, u 2, i 1, c 2, k 1, b 1, r 2, o 3, w 1, n 2, f 2, x 1, j 1, m 1, p 1, v 1, e 4, t 1, h 1

6



題目：比較N個數字



問題描述：輸入一10以下的整數N及N個整數，請依照十進制中各位數字和由大到小排序輸出。如果各位數字和相等則比較數值由大到小排列。例如：9122的各位數字和為 $9+1+2+2=14$ 、3128的各位數字和為 $3+1+2+8=14$ 而5112的各位數字和為 $5+1+1+2=9$ 。所以輸入9122 5112 3128需輸出9122 3128 5112，這是因為 $5112(9) < 3128(14) < 9122(14)$ ，其中又因為3128與9122兩者的各位數字和都是14，所以將數值小的3128放後面。

輸入說明：第一行輸入N，第二行輸入N個整數，數字間用空白隔開， $N < 10$ 。

輸出說明：輸出排序結果。

輸入範例	輸出範例
3 9122 3128 5112	9122 3128 5112

7

WATSE

10.1 簡介與定義



◆結構(structure):

將一些彼此相關的變數結合成相同的名稱，可以含有許多不同資料型別的變數。結構通常會用來定義儲存在檔案裡的記錄(record)。指標和結構可以構成複雜的資料結構，例如鏈結串列、佇列、堆疊和樹等。

結構是一種衍生的資料型別(derived data type)，它是以其他型別的物件來建構的。

```
struct card {
    char *face;
    char *suit;
}; // end struct card
```

WATSE



- ❖ 關鍵字 **struct** 開始一個結構的定義。識別字 **card** 稱為 **結構標籤 (structure tag)**，為結構定義的名稱，它可和保留字 **struct** 一起用來宣告該 **結構型別 (structure type)** 的變數。宣告在結構定義大括號內的變數稱為該結構的 **成員 (member)**。

結構標籤或結構名稱

```
struct card {
    char *face;
    char *suit;
}; // end struct card
```

結構成員



自我參考結構(self-referential structure)



- ❖ 結構不能包含它自己。不過一個指向 **struct employee** 的指標卻可以包含在 **struct employee** 的定義中。
- ❖ 若一個結構含有一個指向相同結構的指標成員，則此結構便稱為一個 **自我參考結構(self-referential structure)**。

```
struct employee2 {
    char firstName[ 20 ];
    char lastName[ 20 ];
    unsigned int age;
    char gender;
    double hourlySalary;
    struct employee2 person; // ERROR
    struct employee2 *ePtr; // pointer
}; // end struct employee2
```

宣告結構型別的變數



- ❖ 結構的定義並**沒有**佔用任何的記憶體空間，它的功能是要建立一種新的資料型別，以供變數宣告之用。結構變數的宣告和其他型別的變數是一樣的。例如

```
struct card aCard, deck[ 52 ], *cardPtr;
```

- ❖ 將aCard宣告為型別struct card的變數，將deck宣告為具有52個struct card型別之元素的陣列，並且將cardPtr宣告為一個指向struct card的指標。



結構標籤名稱



- ❖ 結構標籤名稱是可有可無的。如果某個結構定義沒有標籤名稱，則這種結構型別的變數就一定要宣告在結構定義內，而不能獨立宣告。

```
struct  
{  
    ....  
} var1, var2;
```



可對結構執行的運算



❖ 結構的合法運算只有：

- 設定結構變數的值給另一個相同型別的結構變數
- 取得結構變數的位址 (&)
- 存取結構變數的成員 (見第10.4節)
- 使用**sizeof**運算子來計算結構變數的大小

❖ 思考(考慮儲存空間的變異)

- 結構A == 結構B
- 結構A != 結構B

WATSE

10.3 結構的初始值設定



- ❖ 結構可以像陣列一樣使用初始值串列來設定初始值。要設定初始值，請於定義內的變數名稱後加上一個等號和大括號，於大括號內放入初始值，並以逗號將它們隔開。例如，以下的宣告：

```
struct card aCard = { "Three", "Hearts" };
```

- ❖ 這個宣告產生一個型別**struct card**的變數**aCard**，並且將它的成員**face**設定為**"Three"**，成員**suit**設定為**"Hearts"**。

WATSE

10.4 存取結構的成員



- ❖ 有兩個運算子可以用來存取結構的成員：
 - **結構成員運算子 (.) (structure member operator)** ——也稱為點號運算子 (dot operator)
 - **結構指標運算子 (->) (structure pointer operator)** ——也稱為箭號運算子 (arrow operator)。
 - 結構成員運算子會經由結構變數名稱來存取結構成員。
 - 結構指標運算子是由一個減號 (-) 和一個大於 (>) 緊密結合而成——透過**指向結構的指標 (pointer to the structure)** 來存取結構的成員。

WATSE

```

1 // Fig. 10.2: fig10_02.c
2 // Structure member operator and
3 // structure pointer operator
4 #include <stdio.h>
5
6 // card structure definition
7 struct card {
8     char *face; // define pointer face
9     char *suit; // define pointer suit
10 }; // end structure card
11
12 int main( void )
13 {
14     struct card aCard; // define one struct card variable
15     struct card *cardPtr; // define a pointer to a struct card
16
17     // place strings into aCard
18     aCard.face = "Ace";
19     aCard.suit = "Spades";
20
21     cardPtr = &aCard; // assign address of aCard to cardPtr
22
23     printf( "%s%s\n%s%s\n%s%s\n", aCard.face, " of ", aCard.suit,
24             cardPtr->face, " of ", cardPtr->suit,
25             ( *cardPtr ).face, " of ", ( *cardPtr ).suit );
26 } // end main

```

```

Ace of Spades
Ace of Spades
Ace of Spades

```



10.5 使用結構與函式



- ❖ 結構可以用下列的方式傳給函式：
 - 傳遞個別的結構成員
 - 傳遞整個結構
 - 傳遞指向此結構的指標
- ❖ 當整個結構或個別的結構成員傳給函式時，會以傳值呼叫來加以進行。因此，呼叫函式的結構成員不會被受呼叫的函式更改。
- ❖ 若想以傳參考方式來傳遞結構，程式便需要傳入此結構變數的位址。而結構所組成的陣列則和其他的陣列一樣，會自動以傳參考呼叫的方式來傳遞。

WATSE

10.6 typedef



- ❖ 使用關鍵字 `typedef` 可以為之前定義的資料型別建立別名。結構名稱通常會以 `typedef` 來加以定義，以便建立較短的类型名稱。例如，以下的宣告：

```
typedef struct card Card;
```

- 會將新的型別名稱 `Card` 定義為 `struct card` 型別的別名。
 - 宣告可較簡潔
- ```
struct card cardA;
```
- `Card cardA;`

WATSE

## 練習



- ❖ 請嘗試修改學生成績排序與平均程式，在學生成績資料中加入學生學號，並根據成績對學生做排名
- ❖ 請嘗試修改列出每個字元出現的次數程式，將字元與出現次數當作一筆資料的兩個成員資料
- ❖ 請嘗試修改比較 N 個數字程式，將原數字與數字和當作一筆資料的兩個成員資料