

第七章 陣列

7-1 一維陣列

7-2 二維陣列

7_3 字元陣列與字串

7-4 排序與搜尋

7_5 實例探討

7_6 APCS 觀念題

7_7 APCS 實作題

本書於 3-4 節已介紹若干基本資料型態，像是整數、實數及字元等，本章則針對介紹進階的資料型態陣列(Array)，陣列結構的優點是可以配合迴圈處理大量的數據，請看以下說明。

陣列

前面處理少量的資料，可以個別設定一個變數，但是如果資料龐大，例如，若有資料如下： 12,88,34,45,18,99,22,11,48,77,66

要求極大、極小、平均等運算，若還是使用 11 個變數儲存，那程式將會很冗長，

本章將會介紹另一種資料結構『陣列』，然後配合上一章的 for 或 while 迴圈，

就可以以非常精簡的程式，處理以上大批資料。其次，陣列還有維度之分，分別

是一維、二維、三維等等，那就要看您資料運算的方式，若只有同批資料的關係，

例如，上例就是一維資料陣列就可以。但是，若一個班級所有學生的國、英、數、

自然、社會等 5 科成績，我們不僅要求縱向的各科平均，還要求橫向的每人平

均，那就需要使用二維陣列。請看本章以下說明。

7-1 一維陣列

處理少量的資料，可以為每一筆資料設定一個變數，但若資料數量很多，例如，

若有資料如下：

3,8,4,7,2,9

要求其和、極大、極小呢？是不是指派6 個變數，當然也可以，但是程式會非常冗長，為了改善此一問題，就要使用本節的陣列型態了。因為陣列型態，可以使用『陣列索引』配合for 與while 迴圈而簡化程式，此即為本章重點。例如，以上資料就可宣告陣列如下：

```
int a[7];
```

int 是陣列資料型態，陣列型態可為3_2 節任一型態，a 是陣列變數名稱，變數名稱也應該遵循3_1 節變數命名規則。然後就可依序指派這些資料到陣列，程式如下：

```
a[1]=3;a[2]=8;a[3]=4;a[4]=7;a[5]=2;a[6]=9;
```

中括號內的1,2,3 稱為『陣列索引』，簡稱索引，人類索引編號通常從1 號開始使用，但電腦卻是從0 號開始，所以宣告

```
int a[7]
```

其實是使用0,1,2,3,4,5,6 等7 個位置，索引0 號可用可不用，但有些需要無限循環的情況，就要從0 開始使用。陣列索引從0 開始，有其獨到用法，若配合迴圈與取餘，可以執行一些無限循環的功能。例如，以下程式，在Arduino 控制的應用裡，可以讓PORTA 所連接的LED 每秒顯示一個陣列值，且無限循環。

```
void loop() {  
    PORTA=a[i];  
    delay(1000);  
    i=(i+1) %7;//保障i在0,1,2,3,4,5,6循環  
}
```

陣列的存取

陣列的存取都要靠索引，例如，以下程式可重新指派a 陣列索引3 的值。

```
a[3]=1;
```

以下程式可將a 陣列索引3 的值指派給變數b，但請留意兩者資料型態要相符。

```
int b=a[3];  
int b;b=a[3];
```

陣列宣告與初值設定

單一變數可宣告變數的同時就指派初值，陣列也可以，例如，前面陣列的宣告與

初值指派，程式可以簡化如下：

```
int a[7]={0,3,8,4,7,2,9};
```

本例索引0 不用，也要給予1 個0，那3 才會從索引1 開始放。事實上，陣列宣

告時，您可以不用指派陣列的個數，所以以下程式就可以。

```
int a[]={0,3,8,4,7,2,9};
```

但是陣列的大小給的太小也不行。例如，以下程式就不行：

```
int a[3]={0,3,8,4,7,2,9};
```

其次，給大一點當然可以，但這就浪費了。

```
int a[10]={0,3,8,4,7,2,9};
```

陣列的資料結構的最大優點是，陣列結構可配合迴圈（前面單一變數不行），例

如，以下程式配合陣列的索引，即可以以精簡的程式，處理大批數量的數據。

```
int a[]={1, 2, 3, 4, 5, 6, 7, 8};  
int sum=0;  
for(i=0; i<8; i++) {  
    a[i]=a[i]*2;//每一個都乘以 2  
    sum+=a[i]; //相加每一元素  
}
```

範例 7-1a

假如有資料如下

1, 2, 3, 4, 5, 6, 7, 8

- (1)輸出以上資料。
- (2)計算以上資料的總和。
- (3)請將以上資料乘以 2。
- (4)輸出以上資料。
- (5)請選用適當資料結構儲存以上資料，使得以上資料處理較省事。

執行結果

g702

資料結構

本例雖可用 8 個單獨變數儲存以上資料，但是程式計算其和的程式會非常冗長，若使用一維陣列儲存，那就可配合迴圈，使的程式非常精簡。

程式列印

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int i;
    /* 陣列初始化 */
    int a[]={1, 2, 3, 4, 5, 6, 7, 8};
    int sum=0;
    //簡易輸出
    for(i=0; i<8; i++)
        printf("%d",a[i]);
    printf("\n");
    //輸出索引與值
    for(i=0; i<8; i++)
        printf("a[%d]=%d\n",i,a[i]);
    printf("\n");

    /* 陣列存取及運算 */
    for(i=0; i<8; i++) {
        a[i]=a[i]*2;
        sum+=a[i];
    }
    printf("The sum of array is %d\n", sum);
    printf("After double the array is :\n");
    //輸出索引與值
    for(i=0; i<8; i++)
        printf("a[%d]=%d\n",i,a[i]);
```

```
    printf("\n");
    return 0;
}
```

自我練習

1、程式設計的進階課程有一門是資料結構，資料結構裡最簡單的資料結構是一般變數，其次是陣列、結構、堆疊、佇列、鏈串列、樹狀結構，請同學可使用一般變數重做此題，計算 8 個數字的總和。

2、設有資料如下：

22, 33, 44, 55, 66

(1)請用選用適當資料結構儲存以上資料。

(2)請將以上陣列複製至另一陣列。

(3)將以上兩個陣列內容逐一相加，並放在第三個陣列。

範例 7-1b

請用讀入 5 位學生的國文成績，並可由使用者輸入座號來查詢成績。

執行結果

g703

資料結構

本例也是可使用單獨變數讀入 5 為學生的國文成績，但是程式卻非常冗長，也是選用一維陣列，才可配合迴圈，簡化程式的撰寫。

程式列印

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a[7];
    int c, i;
    /* 成績輸入 */
    for(i=1; i<=5; i++) {
        printf("input the grade of number %d:",i);
        scanf("%d",&a[i]);//留意&
    }
    //輸出索引與值
    for(i=1; i<=5; i++)
        printf("a[%d]=%d",i,a[i]);
    printf("\n");
}
```

```

printf("input a number:");
scanf("%d",&c);//留意&
while (c!=0) {
    printf("a[%d]=%d",c,a[c]);
    printf("input a number:");
    scanf("%d",&c);//留意&
}

return 0;
}

```

自我練習

1. 假設班上共有 12 人，請寫一程式，可以由使用者自行輸入座號，電腦並隨時輸出共有哪些人未到。(提示：可用陣列的值表示到與未到，例如陣列值 1 表到，0 表未到)

```

#include <iostream>
using namespace std;
int main(int argc, char** argv) {
    int n=12;
    int a[n+1];
    int i,j,b;
    // 陣列初始化
    for(i=1; i<=n; i++)
        a[i]=0;
    for(i=1; i<=n; i++){
        cout<<"Check in. Please input your number(1..12):";
        cin>>b;
        a[b]=1;
        for (j=1;j<=n;j++)
            if (!a[j])
                cout<<j<<" ";
        cout<<endl;
    }
    return 0;
}

```

範例 7-1c 設有資料如下

77, 66, 99, 44, 55

請寫一個程式，可以計算平均、最高分及最低分。

執行結果

G704

程式列印

本例為了能使用迴圈來簡化程式，資料結構採用陣列儲存以上資料，程式如下：

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int a[] = { 77, 66, 99, 44, 55 };
    int c, i, max, min, sum;
    max = a[0];
    min = a[0];
    sum = a[0];
    for (i = 1; i <= 4; i++) {
        sum += a[i];
        if (a[i]>max)
            max = a[i];
        if (a[i]<min)
            min = a[i];
    }
    printf( "Average:%d\n", sum / 5);
    printf("max      :%d\n", max );
    printf( "min      :%d\n", min);
    return 0;
}
```

程式說明

1. 求最高分的方法為：
 - (1) 先設第一個人為最高分的擂台主。
 - (2) 第二個人起逐一與擂台主比較，若大於擂台主即取代擂台主。

自我練習

1、設有資料如下：

-1, 0, 3, 4, -5, 8

- (1)統計正數、0、負數的個數。
- (2)請問您的資料結構為何？

2、假設路旁的樹木每 10 公尺種一棵，假設目前樹木存活如下：

001110010011011100

請以陣列儲存以上資料，請問（1）還有幾棵樹存活？（本例 9 顆）（2）連續兩棵樹存活的距離有多少（本例為 50 公尺）。

7_3 字元陣列與字串

C 字串宣告與設定初值

在C 語言中，基本資料型態並無字串資料型態（C++ 才有字串資料型態）。例如：無法使用

```
string a;
```

若要處理字串，則要使用字元陣列或指標（指標不是本書範圍）儲存。在C 語言中，字串是利用簡單的一維字元陣列加上一個字串結束字元'\0'（NULL）來表示。例如：下列敘述是宣告並初始化變數a 的初值為字串"abc"。其次，請注意字串使用雙引號（" "），字元使用單引號（' '），如果誤用則會造成錯誤。

```
char a[]="abc";
```

上式的效果等於

```
char b[]={ 'a', 'b', 'c', '\0'}; /* 將字元單獨放入字元陣列中 */
```

雖然第一種字串宣告方式中的字串"abc" 的長度為3，但是C語言編譯時會自動在字串末端加上一個結束字元（'\0'）表示字串的結束，因此，所宣告的a 字元陣列在記憶體中實際上是占了4 個字元組。請鍵入以下程式，並觀察執行結果。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char *argv[]) {
    char a[]="abc";//會自動放\0,所以長度至少要4
    char b[]={ 'a', 'b', 'c', '\0'};//不會自動放'\0'
    char c[50];
    printf("%c\n",a[0]);//a
    printf("%c\n",a[3]);
    printf("%s\n",a);//abc 留意是s
    printf("%s\n",b);//abc 留意是s
    //字串的複製
    //c=a;//不行，要用strcpy函式
    strcpy(c,a);// #include <string.h>
    printf("%s\n",c);//abc 留意是s
    char d;
    //計算長度，人工複製，要用到字串結束字元\0
    int n=0;int i=0;
    do {
```



```

        d=a[i];
        c[i]=d;
        i++;
    }
    while (d!='\0');
    printf("%s\n",c);//abc 留意是s
    n=i-1;
    printf("the length of c is:%d ",n);//3
    return 0;
}

```

陣列的複製

陣列的複製，並不能使用

```
c=a;
```

一定要使用陣列複製函式strcpy，請參考8-3 節。

範例7_3a假設有資料如下：

姓名	分數
洪國勝	80
張建原	90
張孟庭	88

請以適當的資料結構儲存以上資料，並求其極大值的人名與分數。

輸出結果

G711

程式列印

本例以字元陣列儲存姓名，以變數儲存分數，程式如下：

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>//strcpy()
int main(int argc, char *argv[]) {
    char an[6]="洪國勝";int a=80;//一個中文字佔兩個字元
    char bn[6]="張建原";int b=90;
    char cn[6]="張孟庭";int c=88;
    char maxn[6];int max;
    strcpy(maxn,an);max=a;
    if (b>max){
        strcpy(maxn,bn);
        max=b;
    }
}

```

```
if (c>max){
    strcpy(maxn,cn);
    max=c;
}
printf("Max name is %s ,max score is %d",maxn,max);
return 0;
}
```

補充說明

本例資料僅3筆，一個一個處理尚能應付，若資料增加，則要使用二維陣列，才可配合迴圈，簡化程式，請繼續看以下範例。

此為作者 30 年的教學心得，僅供大家選書參考，更多的精彩內容請購書，這樣作者才有資金繼續研發精彩教材，科技才會進步。