

5-1 一維陣列

重點整理

1. 陣列(array)：一種資料結構，用來儲存多筆**相同資料型態**的資料，每一筆資料稱為元素(element)。
2. 陣列會使用連續的記憶體空間，從第1個元素開始，依序儲存。
3. 宣告陣列的語法：

資料型態 陣列名稱[元素個數]

```
1. int score[10]; /* 宣告一個名稱為 score 的一維陣列，可儲存 10 筆整數資料 */
2. float week[7]; /* 宣告一個名稱為 week 的一維陣列，可儲存 7 筆浮點數資料 */
```

4. 指定陣列元素的初始值

- (1) 在宣告陣列時，同時指定元素個數與元素的初始值：

資料型態 陣列名稱[元素個數] = {初始值 1, 初始值 2, ...};

- (2) 在宣告陣列時，不指定元素個數，但指定各元素的初始值，編譯器會依大括號內的初始值數量，自動判斷元素個數：

資料型態 陣列名稱[] = {值 1, 值 2, ...};

```
1. int K[5] = {12, 34, 56, 77, 51}; /* 宣告一個名稱為 K 的整數陣列，指定元素個數與元素的初始值 */
2. float F[] = {1.2, 3, 8, 7.3, 9, 5, 10.4}; /* 宣告一個名稱為 F 的浮點數陣列，指定初始值，不指定
   元素個數 */
```

5. 元素的存取方式：

- (1) 使用索引(index)的方式存取陣列元素，**索引的最小值為 0**，最大值為**元素個數-1**

- (2) 陣列的第一個元素，其索引值為 0

- (3) 指定陣列元素的內容

陣列名稱[索引值] = 值;

- (4) 將陣列元素的內容指定給其它變數

變數 = 陣列名稱[索引值];

6. 陣列通常會搭配迴圈使用，以迴圈的控制變數為索引值，循序走訪每一個元素。

```
1. int i, A[3];
2. for (i=0; i<3; i++){
3.     A[i]=i;                      /* 指定陣列 A[] 的元素分別為 0、1、2 */
4. }
5. for (i=0; i<3; i++){
6.     printf("%d", A[i]);        /* 輸出陣列 A[] 的每一元素，顯示 012 */
7. }
```

7. 在 C 語言中，沒有「字串」資料型態，而是使用字元陣列來表示字串，且在字元陣列的最後，會自動加上一個**結束字元'\0'**。
8. 字元陣列的名稱，即為該字串的內容。下列範例皆是宣告字元陣列的方式，每個字元陣列都包含 6 個元素。

```

1. char name1[6]={'m','a','r','i','o','\0'};
2. char name2[6]="mario";
3. char name3[]={ 'm', 'a', 'r', 'i', 'o', '\0' };
4. char name4[]="mario";
5. printf("%s\n", name1); /* printf() 使用格式字元「%s」輸出字串，印出 mario */
6. printf("%s\n", name2); /* 印出 mario */
7. printf("%s\n", name3); /* 印出 mario */
8. printf("%s\n", name4); /* 印出 mario */

```

9. C++ 語言有提供字串 **string** 資料型別，使用前需引用 **<string>** 函式庫。

```

1. #include <string> // 引用函式庫標頭檔 <string>
2. :
3. string name = "mario"; /* 變數 name 為 string 資料型態 */
4. cout << name << endl; /* 使用 cout 物件輸出變數內容 */

```

10. 使用 **sizeof()** 函式，可以取得陣列使用的記憶體空間大小，

```

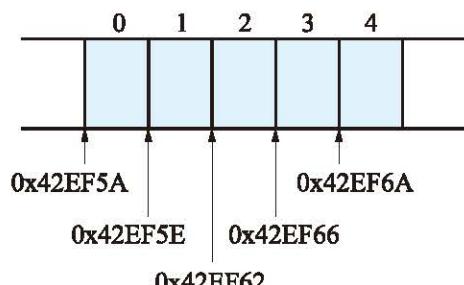
1. int A[] = {1,2,3,4,5};
2. char S[] = "Happy";
3.
4. printf("%d\n", sizeof(A)); /* int 型態為 4Byte，5 個元素共 20Byte，輸出 20 */
5. printf("%d\n", sizeof(S)); /* char 型態為 1Byte，6 個元素（需考慮結束字元'\0'）共 6Byte，輸出 6 */

```

01 範例

在 C 語言中宣告一個 int 資料型態的整數陣列，該陣列有 5 個元素，已知第 1 個元素的記憶體位址為 0x42EF5A，則最後 1 個元素的位址為何？

解 int 資料型態的大小為 4Byte，因此陣列的每個元素大小皆為 4Byte，所以第 5 個元素的位址為 $0x42EF5A + (5-1) * 4 = 0x42EF6A$



02 範例

表頭題

小明的第 1 次段考成績如下，請使用一維陣列結構，幫他寫一支計算平均成績的程式。

國文	英文	數學	基本電學	程式設計實習
85	80	90	96	88

解

程式碼：

```

1. #include <stdio.h>
2.
3. int main(){
4.     int score[5]={85,80,90,96,88};
5.     float avg;
6.     for (int i=0;i<5;i++)
7.         avg=avg+score[i];
8.     avg=avg/5;
9.     printf("平均成績:%f", avg);
10. }
```

程式說明：

行數	說明
4	宣告整數陣列 score，指定 5 個元素的初始值。
6,7	使用迴圈走訪陣列，將各元素加總至變數 avg。
8	計算平均值。

03 範例

素養題

某公司的每月營業額如下表，請設計一程式，執行該程式會輸出營業額最高的月份以及該月的營業額。

月份	1	2	3	4	5	6	7	8	9	10	11	12
營業額 (仟萬)	900	850	925	1012	936	998	1019	899	986	978	1002	1011

解

程式碼：

```

1. #include <stdio.h>
2.
3. int main(){
4.     int money[12]={900,850,925,1012,936,998,1019,899,986,978,1002,1011};
5.     int max=0;
6.     for (int i=1;i<12;i++){
7.         if(money[i]>money[max])
8.             max=i;
9.     }
10.    printf("營業額最高的月份 : %d 月\n", max+1);
11.    printf("營業額 : %d 仟萬元",money[max]);
12. }
```

程式說明：

行數	說明
4	宣告整數陣列 money，記錄每月的營業額，money[0]為 1 月的營業額，money[1]為 2 月的營業額，以此類推。
5	(1) 變數 max 記錄陣列中最大值的索引。 (2) 初始假設 1 月的營業額最高（陣列的第一個元素，索引值為 0），故指定 max 的初始值為 0。
6 ~ 9	從 money[1]開始，依序和 money[max]比較，使 max 永遠為目前最大值的索引，此迴圈結束後，max 的值為 6。
10 ~ 11	輸出營業額最高的月份和該月營業額。

04 範例

素養題

某期的樂透開獎號碼是 5, 17, 34, 19, 2, 29，請將此 6 個號碼由小到大排序。

解

程式碼：

```

1. #include <stdio.h>
2.
3. int main(){
4.     int lottery[]={5, 17, 34, 19, 2, 29};
5.     int i,j,tmp;
6.     for (i=4; i>=0; i--){
7.         for (j=0; j<=i; j++){
8.             if(lottery[j]>lottery[j+1]){
9.                 tmp=lottery[j];
10.                lottery[j]=lottery[j+1];
11.                lottery[j+1]=tmp;
12.            }
13.        }
14.    }
15.    printf("本期樂透開獎號碼：");
16.    for(i=0;i<6;i++)
17.        printf("%d",lottery[i]);
18. }
```

程式說明：

行數	說明
4	宣告整數陣列 lottery，記錄 6 個樂透號碼。
6 ~ 8	使用泡沫排序法，將陣列元素兩兩比較，使最大值往陣列右側移動，陣列內容的變化情形如下圖，

行數

說明

(1) 當 $i == 4$ 時，

i	j	
4	0	比較，不交換
	1	比較，不交換
2		比較，交換 i
3		比較，交換
4		比較，交換

行數

說明

(2) 當 $i == 3$ 時，(3) 當 $i = 2, 1, 0$ 時的程式執行步驟，請讀者自行練習

9 ~ 11

lottery[j]和lottery[j+1]交換

•立即練習(B) 1. 對於陣列的描述，下列何者錯誤？

- (A)陣列是一種存放資料的結構
- (B)陣列在宣告時，需同時指定元素個數與元素的初始值
- (C)陣列的索引值最小為 0
- (D)陣列的每一個元素，必須是相同的資料型態。

(D) 2. `int abc[4] = {1, 3, 5, 7};` 何者錯誤？

- (A)這是一個整數陣列
- (B)此陣列可儲存 4 個元素
- (C)`abc[3]`的值為 7
- (D)陣列的索引值從 1 開始。

(A) 3. 在 C 語言中，要依序讀取陣列中的每一個元素，哪一種語法最適合？

- (A)for
- (B)if
- (C)switch
- (D)break。

(B) 4. 陳老師打算寫一支程式來處理全班的基本電學平時成績，請問使用下列哪一種資料結構最適合？ (A)連結串列 (B)陣列 (C)雜湊 (D)堆疊。

- (C) 5. 在 C 語言中宣告一個 4 個元素的 int 型態的陣列，會佔用多少記憶體空間？
 (A)4Byte (B)8Byte (C)16Byte (D)32Byte。
- (B) 6. 某一維 int 型態的整數陣列有 10 個元素，第 1 個元素的位址為 0x1357，則最後一個元素的位址為何？ (A)0x137A (B)0x137B (C)0x137C (D)0x137D。
- (B) 7. C 語言中使用字元陣列來表現字串，「char name[]="John";」這行敘述中，字元陣列 name 的元素有幾個？ (A)6 (B)5 (C)4 (D)3。
- (C) 8. 承上題，執行 strlen(name); 會得到何值？ (A)6 (B)5 (C)4 (D)3。
- (A) 9. 下列程式片段執行後，score[5]的內容為何？ (A)5 (B)6 (C)7 (D)8。

```
int score[10];
for (int i=0; i<10; i++){
    score[i] = 0;
    score[i] = score[i] + i;
}
```

- (C) 10. 承上題，已知 int 型態的大小為 4byte，則執行 sizeof(score); 的結果為何？
 (A)10 (B)20 (C)40 (D)80。
- (B) 11. C 語言使用字元陣列來表示字串，小美要使用 printf() 輸出函式將某字串輸出，需使用哪一個格式字元？ (A)%d (B)%s (C)%f (D)%i。
- (D) 12. 下列程式片段執行後，data[9]的值為何？

(A)9 (B)2 (C)10 (D)18。

```
int data[10];
for (int i=0; i<10; i++){
    data[i]=i*2;
}
```

- (B) 13. 下列程式片段執行後，sum 的值為何？
 (A)5 (B)25 (C)55 (D)110。

```
int sum = 0;
int num[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

for (int i=0; i<10; i++){
    if(i%2==0)
        sum=sum+num[i];
}
```

(A) 14. 下列程式片段執行後，A[9]的值為何？ (A)34 (B)21 (C)13 (D)55。

```
int A[10];
A[0]=0;
A[1]=1;
for (int i=2; i<10; i++)
    A[i]=A[i-1]+A[i-2];
```

(A) 15. 下列程式片段執行後的輸出為何？

(A)9 9 9 9 9 (B)10 10 10 10 10 (C)5 5 5 5 5 (D)0 1 2 3 4。

```
int t[10];
for (int i=0; i<10; i++){
    t[i] = i;
}
for (int i=0; i<5; i++){
    printf("%d", t[i] + t[9-i]);
}
```

(C) 16. 下列程式片段執行後的輸出為何？ (A)0,9 (B)12,82 (C)7,17 (D)17,7。

```
int data[] = {9,51,41,87,46,93,16,17,55,12,34,67,82};
int p,t;
t=17;
for(int i=0; i<13; i++){
    if (data[i] == t){
        p=i;
        break;
    }
}
printf("%d,%d" , p, data[p]);
```

(D) 17. 下列關於陣列 var 的宣告方式，何者正確？

(A)long var[1.23]; (B)long var[] ={1.23};
(C)float var[1.23]; (D)float var[] ={1.23};。

(A) 18. 下列關於 C 語言陣列之敘述，何者正確？

(A)int var[3]是指宣告了可以存放 3 個整數變數的陣列
(B)int var[1]是指宣告了可以存放 2 個整數變數的陣列
(C)陣列索引的預設起始值為 1
(D)陣列在使用前不需事先宣告。

(B) 19. 請問下列 C 程式片段執行後的結果為何？

(A)10 (B)11 (C)12 (D)13。

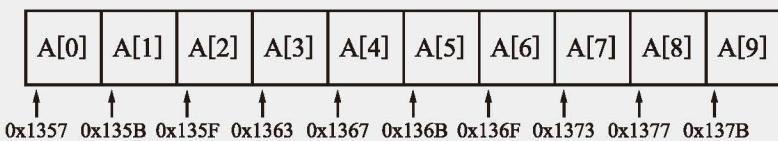
```
int A[5]={1, 2, 3, 4, 5};
printf("%d", A[2]+A[A[0]+A[1]]+A[3]);
```

答案 打★表示有詳解

★1.(B) 2.(D) 3.(A) 4.(B) 5.(C) ★6.(B) ★7.(B) ★8.(C) 9.(A) ★10.(C)
 11.(B) 12.(D) 13.(B) 14.(A) 15.(A) ★16.(C) 17.(D) ★18.(A) 19.(B)

解析

1. 陣列宣告時，可不指定初始值。
6. $0x1357 + (10-1)*4 = 0x137B$ ，陣列在記憶體內的位置如下，



7. 包含結束字元' \0 '。
8. 字串函式 `strlen()` 可以取到字串長度(不包含結束字元' \0 ')。
10. `sizeof`(陣列名稱)會回傳陣列佔用的記憶體空間。
16. 此程式在搜尋陣列中的特定元素。
18. (B) `int var[1]` 告訴可以存放 1 個整數的陣列
 (C) 陣列索引的起始值為 0
 (D) 陣列在使用前需先宣告

5-2 二維陣列與多維陣列

重點整理

1. 宣告二維陣列的語法：

資料型態 陣列名稱[列數][行數]

```
1. int ball[3][4]; /* 宣告一個 3 列 4 行的二維陣列 */
```

2. 宣告二維陣列的同時，除了指定列數與行數外，也可以同時指定陣列元素的初始值。

```
1. int ball[3][4] = {{0,1,2,3},  

2.                 {4,5,6,7},  

3.                 {8,9,10,11},  

4.             }; /* 宣告一個 3 列 4 行的二維陣列，同時指定元素初始值 */
```

另外也可以在宣告時不指定列數和行數，但指定各元素的初始值，由編譯器決定陣列的大小。

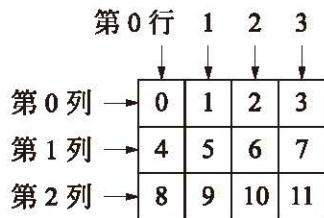
```
1. int ball[][] = {{0,1,2,3},  

2.                 {4,5,6,7},  

3.                 {8,9,10,11},  

4.             }; /* 不指定列行數，但指定元素初始值 */
```

上述兩個程式片段執行後，陣列 ball[3][4]的內容相同，下圖可以具體的呈現二維陣列中各元素間的相對位置：



3. 二維陣列在存取時，需同時使用列索引和行索引，

(1) 指定陣列元素的內容

陣列名稱[列索引] [行索引] = 值;

(2) 將陣列元素的內容指定給其它變數

變數 = 陣列名稱[列索引] [行索引];

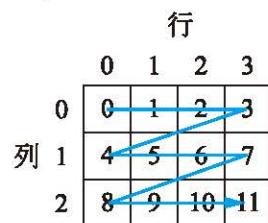
```

1. int ball[3][4] = {{0,1,2,3},
2.                 {4,5,6,7},
3.                 {8,9,10,11},
4.             }; /* 宣告一個 3 列 4 行的二維陣列，同時指定元素初始值 */
5.
6. printf("%d\n", ball[1][2]); /* 第 1 列第 2 行，顯示 6 */
7. ball[1][2]= 99; /* 將第 1 列第 2 行的值，指定為 99 */

```

4. 在程式中，使用巢狀迴圈走訪二維陣列的方式：

(1) 以列為主 (Row-major) (行優先)：



```

1. for (int i=0;i<3;i++){      /* 第 i 列 */
2.     for (int j=0;j<4;j++){  /* 第 j 行 */
3.         printf("%d,\n", ball[i][j]); /* 輸出 0,1,2,3,4,5,6,7,8,9,10,11 */
4.     }
5. }

```

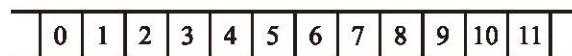
(2) 以行爲主 (Column-major) (列優先) :

		行			
		0	1	2	3
列	0	0	1	2	3
	1	4	5	6	7
2	8	9	10	11	

```

1. for (int j=0;j<4;j++){ /* 第 j 行 */
2.     for (int i=0;i<3;i++){ /* 第 i 列 */
3.         printf("%d\n", ball[i][j]); /* 輸出 0,4,8,1,5,9,2,6,10,3,7,11 */
4.     }
5. }
```

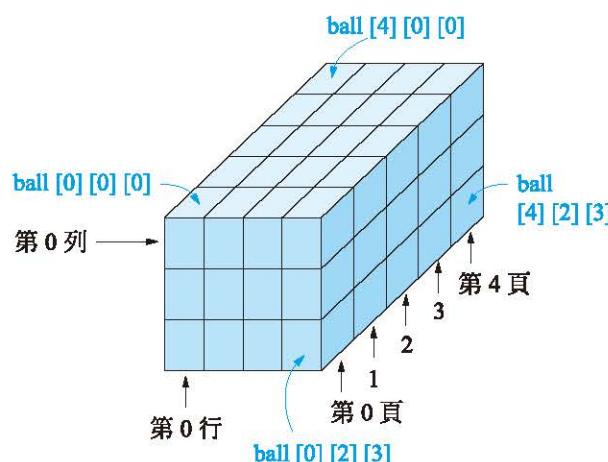
5. 實際上，C/C++語言採用以列爲主 (Row-major) (行優先) 的方式，將二維陣列的元素存放在連續的記憶體位置，以上述的陣列 `ball[3][4]` 為例，各元素在記憶體中的儲存位置如下圖，因此在撰寫程式時，使用以列爲主 (行優先) 的方式走訪陣列，執行效率較佳。



6. 三維陣列：將多個相同資料型態與大小的二維陣列疊在一起。

```
1. int ball[5][3][4];
```

上面這行程式碼，宣告一個 5 頁 3 列 4 行的 3 維陣列，如下圖所示。



05 範例

素養題

2020 中華職棒總冠軍戰打滿 7 場，獅、象 2 隊各場的得分如下，請幫聯盟撰寫一程式，計算兩隊 7 場的總得分。

	第 1 場	第 2 場	第 3 場	第 4 場	第 5 場	第 6 場	第 7 場
獅	4	1	1	3	6	12	7
象	2	9	5	6	0	1	4

解

程式碼：

```

1. #include <stdio.h>
2.
3. int main(){
4.     int ball[2][8]={{0,4,1,1,3,6,12,7},{0,2,9,5,6,0,1,4}};
5.
6.     for(int i=0;i<2;i++){
7.         for(int j=1;j<8;j++){
8.             ball[i][0]=ball[i][0]+ball[i][j];
9.         }
10.    }
11.
12.    printf("統一獅總分 : %d\n", ball[0][0]);
13.    printf("兄弟象總分 : %d\n", ball[1][0]);
14. }
```

程式說明：

行數	說明
4	宣告一個 2 列 8 行的二維陣列，利用每列的第一個元素(即 ball[0][0] 和 ball[1][0])記錄總分。
6 ~ 10	採用以列為主(Row-major)(行優先)的方式循序讀取陣列，將每列的第 2 到 8 個元素加總。

06 範例

素養題

某選秀節目有 5 位參賽者進入決賽，由評審老師針對不同的表演項目進行評分，請幫製作單位撰寫程式，計算參賽者及表演項目的平均得分。

	歌唱技巧	台風	特殊才藝	平均分數
周杰倫	85	87	88	
張會妹	86	88	87	
蔡依靈	87	86	90	
田馥真	80	84	86	
吳青峰	90	89	91	
平均分數				

解

程式碼：

```

1. #include <stdio.h>
2.
3. int main(){
4.     float score[6][4]={{85,87,88,0},
5.                         {86,88,87,0},
6.                         {87,86,90,0},
7.                         {80,84,86,0},
8.                         {90,89,91,0},
9.                         {0,0,0,0}
10.    };
11.    /* 以列為主 */
12.    for(int i=0;i<5;i++){      /* 第 i 列 */
13.        for(int j=0;j<3;j++)   /* 第 j 行 */
14.            score[i][3]=score[i][3]+score[i][j];
15.        score[i][3]=score[i][3]/3;
16.    }
17.    /* 以行為主 */
18.    for(int j=0;j<3;j++){      /* 第 j 行 */
19.        for(int i=0;i<5;i++){  /* 第 i 列 */
20.            score[5][j]=score[5][j]+score[i][j];
21.        }
22.        score[5][j]=score[5][j]/5;
23.    }
24. }
```

程式說明：

行數	說明
4 ~ 10	宣告一個 6 列 4 行的二維陣列，利用最後一列與最後一行，分別記錄參賽者及表演項目的平均得分。
12 ~ 16	計算每位參賽者的平均分數，以列為主(Row-major)(行優先)的方式循序讀取陣列，將每列的第 1 到 3 個元素加總後再取平均。
18 ~ 23	計算每個表演項的平均分數，採用以行為主(Column-major)(列優先)的方式循序讀取陣列，將每行的第 1 到 5 個元素加總後再取平均。

• 立即練習

- (B) 1. 宣告一個 4 列 5 行的二維陣列，則此陣列的元素個素有幾個？
 (A)30 (B)20 (C)50 (D)60。
- (D) 2. 宣告一個 3 維陣列， $A[3][4][5]$ ，則此陣列的元素個素有幾個？
 (A)30 (B)20 (C)50 (D)60。
- (B) 3. 宣告一二維陣列 $\text{int } Y[3][4]=\{\{45,5,43,21\},\{12,37,21,48\},\{84,91,23,55\}\}$ ；下列哪一個選項可以取得元素 91？ (A) $Y[3][2]$ (B) $Y[2][1]$ (C) $Y[1][2]$ (D) $Y[2][3]$ 。
- (C) 4. 承上題， $Y[1][3]$ 的值為何？ (A)43 (B)21 (C)48 (D)23。
- (B) 5. 承上題，C 語言使用以列為主(Row-major)的方式將二維陣列的資料循序存放在記憶體中，若 $Y[0][0]$ 放在記憶體的第一個位置，則 $Y[2][1]$ 會放在記憶體的第幾個位置？
 (A)9 (B)10 (C)5 (D)6。
- (A) 6. 執行以下的程式片段後， $abc[1][6]$ 的內容為何？ (A)7 (B)6 (C)5 (D)4。

```
int abc[2][10];
int i,j;
for (i=0;i<2;i++){
    for (j=0;j<10;j++){
        abc[i][j] = i+j;
    }
}
```
- (D) 7. 有關 C 語言程式陣列宣告，以下何者錯誤？
 (A) $\text{int } a[2][3];$ (B) $\text{int } b[] [3]=\{0,1,2,3\};$
 (C) $\text{int } c[100][100]=\{0\};$ (D) $\text{int } d[3][]=\{\{1,2\},\{1,2,3\},\{1,2,3,4\}\};$ 。
- (A) 8. 大部分程式語言都是以列為主的方式儲存陣列。在一個 8×4 的陣列(array) A 裡，若每個元素需要兩單位的記憶體大小，且若 $A[0][0]$ 的記憶體位址為 108 (十進制表示)，則 $A[1][2]$ 的記憶體位址為何？ (A)120 (B)124 (C)128 (D)以上皆非。

(D) 9. 下列程式片段執行後， $D[2][1][0]$ 的值為何？ (A)0 (B)12 (C)121 (D)21。

```
int D[3][3][3];
for (int i=0;i<3;i++){
    for (int j=0;j<3;j++){
        for (int k=0;k<3;k++){
            D[i][j][k]=i*9+j*3+k;
        }
    }
}
```

答案 打★表示有詳解

1.(B) 2.(D) 3.(B) 4.(C) ★5.(B) 6.(A) ★7.(D) ★8.(A) 9.(D)

解析

5. 以列為主的排列方式，陣列各元素在記憶體內的位置如下：

[0][0]	[0][1]	[0][2]	[0][3]	[1][0]	[1][1]	[1][2]	[1][3]	[2][0]	[2][1]	[2][2]	[2][3]
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

7. 宣告二維陣列時，列數可省略，行數不可省略。

8. $108 + [(1-0) \times 4 + (2-0)] \times 2 = 120$ ，陣列 $A[][]$ 在記憶體內的位置如下，

A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[1][0]	A[1][1]	A[1][2]	A[1][3]	A[2][0]	A[2][1]	A[2][2]	A[2][3]	...
↑ 108	↑ 110	↑ 112	↑ 114	↑ 116	↑ 118	↑ 120						

5-3 指標

重點整理

- 在記憶體內，以 1 Byte 作為一個基本的儲存單位，每個儲存單位都有一個編號，稱之為位址，通常用 16 進位表示。
- 程式執行時，會依據變數的資料型態，配置適當大小的記憶體空間，例如 int 型態的大小為 4 Byte，作業系統會在記憶體中配置 4 個儲存單位。

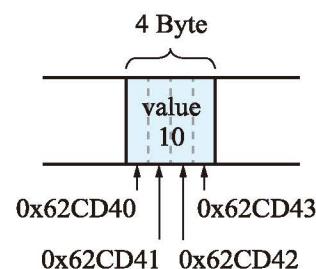
1. `int value = 10;`

這行程式碼，會佔用記憶體 4 個儲存單位，其內容為 10，使用變數名稱 `value` 存取內容（變數值），記憶體內容如右圖，

- 透過變數存取記憶體特定位置的資料，優點是不會不小心改到記憶體其它位置的資料，造成不可預期的錯誤，但缺點是無法得知資料在記憶體中的真正位址，較無彈性。
- 在 C/C++ 語言中，使用「指標」表示資料所在的記憶體位置，換言之，**指標就是「記憶體位址」**，要取得某個變數的記憶體位址，需使用取址運算子「&」，語法如下：

&變數

2. `printf("%p", &value); /* 變數 value 的記憶體位址：0062CD40 */`



5. 要記錄記憶體位址，必需使用特殊的變數，稱為「指標變數(Point Variable)」，就像整數變數記錄整數一樣，**指標變數記錄的是指標(位址)**。
6. 指標變數的宣告語法：

資料型態 *指標變數名稱

或

資料型態* 指標變數名稱

宣告指標變數時的資料型態，指的並非是指標變數的資料型態(指標變數只能存放位址)，而是該**指標指向的變數的資料型態**。

3. int *ptr; /*宣告一個指標變數 ptr，ptr 只能指向整數變數 */

7. 將變數 value 的位址指定給指標變數 ptr，亦即，**指標變數 ptr 指向變數 value**。

4. ptr = &value; /* 將 value 的位址指定給指標變數 ptr */

8. 指標變數的值必須是記憶體位址，不可以是常數，若暫時不指定位址，最好先將指標變數指定為 NULL。
9. 要存取指標變數指向的記憶體位址的內容(即指向的變數值)，需使用取值運算元「*」，語法如下：

*指標變數

5. printf("%d", *ptr); / 輸出指標變數 ptr 指向的記憶體位址的內容：10 */

10. 以下程式碼說明變數 value 和指標變數 ptr 之間的關係，並整理成下表，

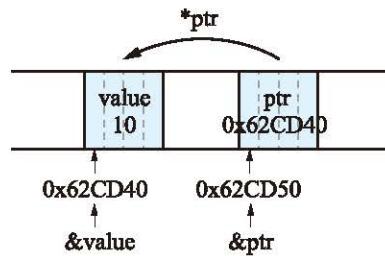
```

1. int value=10;      /* 宣告一個整數變數 value */
2. int *ptr;          /* 宣告一個指標變數 ptr */
3. ptr = &value;       /* 將變數 value 的位址，指定給指標變數 ptr (ptr 指向 value) */
4. printf("%d\n", value); /* 變數 value 的值 */
5. printf("%p\n", &value); /* 變數 value 的位址 */
6. printf("%p\n", ptr);   /* 指標變數 ptr 的值 */
7. printf("%d\n", *ptr);  /* 指標變數 ptr 指向的記憶體位址內容 */
8. printf("%p\n", &ptr);  /* 指標變數 ptr 的位址 */

```

	值	說明
value	10	變數 value 的值
&value	0062CD40	變數 value 的記憶體位址
ptr	0062CD40	指標變數 ptr 的值，即變數 value 的位址
*ptr	10	指標變數 ptr 指向的記憶體位址的內容，即變數 value 的值
&ptr	0062CD50	指標變數 ptr 本身的記憶體位址(指標變數也是變數，同樣會被儲存在記憶體中的某個位置)

記憶體內容如下圖：



11. 指標變數只能進行+、-、++、--四種運算，運算的結果會改變指向的記憶體位址，每次變化基本單位是指向資料的大小(以上述程式碼的整數指標變數 `ptr` 為例，執行 `ptr++` 後，因整數大小為 4 Byte，`ptr` 的內容會從 `0x62CD40` 變為 `0x62CD44`)。
12. 陣列名稱又被用來表示陣列第 1 個元素的位址，所以陣列名稱可視為是指向該陣列的**陣列指標變數**。
13. 陣列指標變數只能指向陣列的第 1 個元素，不可以指向其它變數。
14. 下列程式碼說明陣列名稱與陣列元素間的關係，並整理成下表，

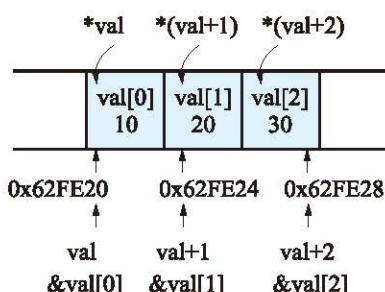
```

1. int val[3]={10,20,30};
2. printf("%p\n", &val[0]); /* 陣列第 1 個元素的記憶體位址 */
3. printf("%p\n", val); /* 陣列名稱是第 1 個元素的記憶體位址 */
4. printf("%p\n", val+1); /* 第 2 個元素的記憶體位址 */
5. printf("%p\n", val+2); /* 第 3 個元素的記憶體位址 */
6. printf("%d\n", *val); /* 對陣列名稱取值，即第 1 個元素的值 */
7. printf("%d\n", *(val+1)); /* 第 2 個元素的值 */
8. printf("%d\n", *(val+2)); /* 第 3 個元素的值 */

```

	值	說明
<code>&val[0]</code>	0062FE20	第 1 個元素的記憶體位址
<code>val</code>	0062FE20	陣列的名稱，代表該陣列第 1 個元素的記憶體位址
<code>val+1</code>	0062FE24	第 2 個元素的記憶體位址
<code>val+2</code>	0062FE28	第 3 個元素的記憶體位址
<code>*val</code>	10	陣列第 1 個元素的值，即 <code>val[0]</code>
<code>*(val+1)</code>	20	陣列第 2 個元素的值，即 <code>val[1]</code>
<code>*(val+2)</code>	30	陣列第 3 個元素的值，即 <code>val[2]</code>

記憶體內容如下圖：



15. C 語言除了使用字元陣列表示字串，也可以將字串直接指定給字元指標變數，以下程式碼說明如何使用字元陣列或字元指標變數處理字串。

```

1. char strA[]="SuperMario"; /* 壓告字元陣列 strA[]，陣列名稱 strA 為第 1 個字元 'S' 的位址 */
2. char *strP="PokemonGo"; /* 壓告字元指標變數 strP，strP 的值為第 1 個字元 'P' 的位址 */
3.
4. printf("%s\n", strA); /* 格式字元「%s」，輸出字串 SuperMario */
5. printf("%s\n", strA+5); /* 格式字元「%s」，輸出字串 Mario */
6. printf("%s\n", strP); /* 格式字元「%s」，輸出字串 PokemonGo */
7. printf("%s\n", strP+7); /* 格式字元「%s」，輸出字串 Go */
8.
9. printf("%c\n", *strA); /* 格式字元「%c」，輸出 strA 指向的字元 S */
10. printf("%c\n", *(strA+5)); /* 格式字元「%c」，輸出 strA+5 指向的字元 M */
11. printf("%c\n", *strP); /* 格式字元「%c」，輸出 strP 指向的字元 P */
12. printf("%c\n", *(strP+7)); /* 格式字元「%c」，輸出 strP+7 指向的字元 G */

```

07 範例

執行以下之程式碼後，a、b、*p、*q 的值分別為何？

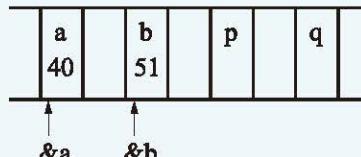
```

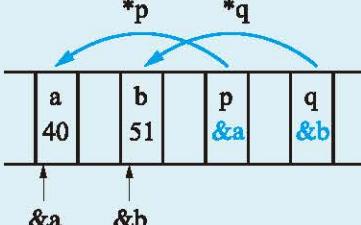
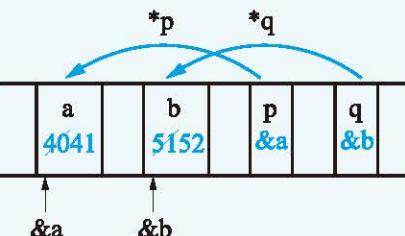
1. #include <stdio.h>
2. main()
3. {
4.     int a=40, b=51;
5.     int *p, *q;
6.     p = &a;
7.     q = &b;
8.     (*p)++;
9.     (*q)++;
10.    q = p;
11.    printf("%d,%d,%d,%d", a, b, *p, *q);
12. }

```

解

行數	說明
4、5	宣告整數變數 a、b，宣告整數指標變數 p、q。



行數	說明
6、7	將變數 a 的位址指定給指標變數 p，即 p 指向 a。 將變數 b 的位址指定給指標變數 q，即 q 指向 b。 
8、9	使用取值運算子「*」對指向的變數值進行++運算。 
10	使 p 和 q 都指向整數變數 a
11	輸出 41,52,41,41

08 範例

承範例 2，使用陣列索引的方式存取陣列元素，請使用陣列指標改寫該範例。

解

程式碼：

```

1. #include <stdio.h>
2.
3. int main(){
4.     int score[5]={85,80,90,96,88};
5.     float avg;
6.     for (int i=0;i<5;i++)
7.         avg=avg+*(score+i);
8.     avg=avg/5;
9.     printf("平均成績:%f", avg);
10. }
```

程式說明：

行數	說明
6~7	利用陣列名稱 score 來走訪陣列，使用取值運算子「*」取出元素後，加總至變數 avg。

• 立即練習

- (D) 1. 關於 C 語言指標的描述，下列何者錯誤？
- (A)指標變數用來存放記憶體位址 (B)「*」是取值運算子
 (C)「&」是取址運算子 (D)「int *ptr;」int 表示 ptr 的值需是整數。
- (D) 2. 在 C 語言中宣告一個指標變數 ptr，哪一個是錯誤的語法？
- (A)int *ptr; (B)int* ptr; (C)double *ptr; (D)int *ptr=5;。
- (C) 3. 宣告一個指標變數 p 和一個整數變數 k，下列哪一個敘述可以讓指標 p 指向變數 k？
- (A)p=k; (B)p=*k; (C)p=&k; (D)p=&&k。
- (B) 4. 宣告一指標變數後，若暫時不指定指向的位址，可以先設定為何來避免造成不可預期的錯誤？ (A)ZERO (B)NULL (C)-1 (D)EMPTY。
- (B) 5. 要取出指標變數指向的變數值，需使用哪一個運算元？
- (A)& (B)* (C)# (D)%。
- (C) 6. 關於 C/C++ 語言的描述，下列何者錯誤？
- (A)指標變數不可以設定為常數
 (B)陣列名稱可以用來表示其第 1 個元素的指標
 (C)指標變數不可以進行++運算
 (D)陣列的索引值最小值為 0。
- (A) 7. 下列程式片段執行後的輸出為何？ (A)100 (B)90 (C)80 (D)70。
- ```
int data=100;
int *pData;
pData = &data;
printf("%d\n", *pData);
```
- (B) 8. 下列程式片段執行後的輸出為何？ (A)55,55   (B)66,66   (C)77, 77   (D)88,88。
- ```
int k = 66;
int *p;
p = &k;
printf("%d, %d", *p, k);
```

- (A) 9. 下列程式片段執行後的輸出為何？ (A)34,34 (B)66,66 (C)34,66 (D)66,34。

```
int k = 66;
int *p;
p = &k;
*p = 100 - *p;
printf("%d, %d", *p, k);
```