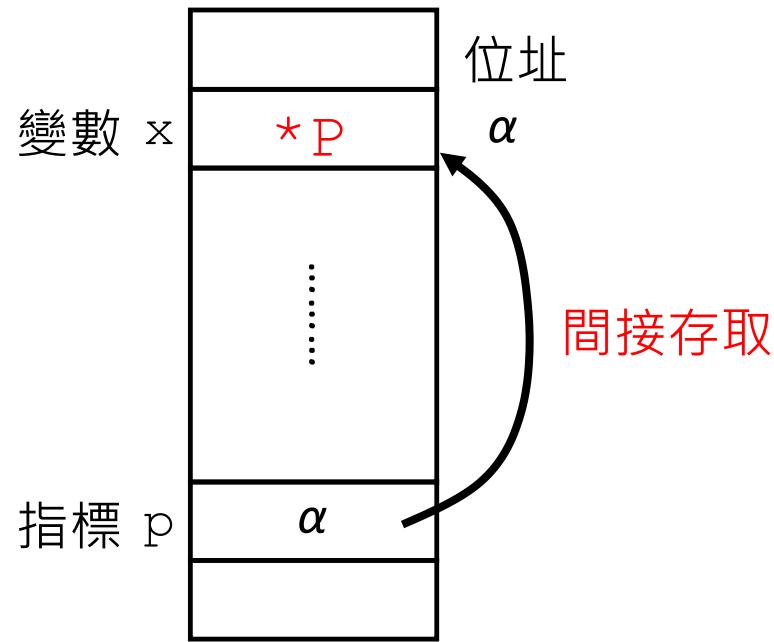
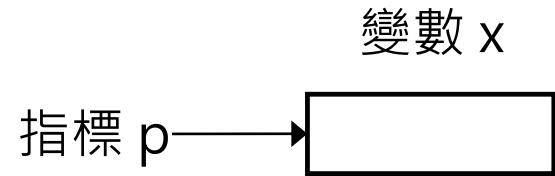


陣列與指標

鍾宜玲

指標POINTER的定義

- 儲存變數的位址
- 若指標 p 儲存變數 x 的位址，稱為「指標 p 指向變數 x 」
- $*p$ ：間接存取，透過指標 p 存取此記憶體的資料。



以C語言實作指標

■ 語法：

資料型態 *指標變數；

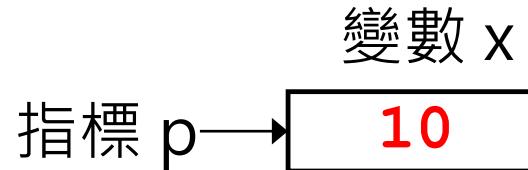
例如：int *p;

■ 將某變數之位址存入指標中，利用&符號：

int x, *p;

p=&x; // &x 表示x變數的位址

*p=10; // 相當於 x=10;



練習(1)：寫出執行結果

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a=10, b=20;
    int *p1, *p2;
    p1=&a;
    p2=&b;
    printf("a=%d\n", *p1);
    printf("b=%d\n", *p2);
    printf("a的位址=%p\n", p1);
    printf("b的位址=%p\n", p2);
    *p1+=5;
    *p2=a+b;
    printf("a=%d\n", *p1);
    printf("b=%d\n", *p2);
    system("pause");
    return 0;
}
```

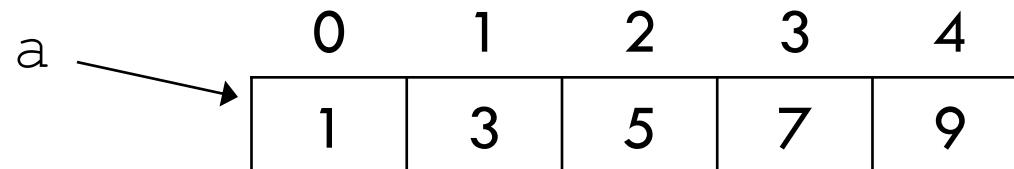
指標與陣列



當宣告整數陣列如下：

```
int a[] = {1, 3, 5, 7, 9};
```

1. 陣列名稱 a 為靜態指標 (指標常數)
2. 即永遠指向陣列的起始位址
3. 不能在程式中改變指標 a 的內容



練習(2)：執行並寫出結果

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a[]={1, 3, 5, 7, 9}, i, *p;
    p = a;
    for(i=0; i<5; i++) {
        printf("a[%d]=%d\n", i, *p);
        p++;
    }
    system("pause");
    return 0;
}
```

指標的運算



```
int a[] = {1, 3, 5, 7, 9};
```

```
int *p; //動態指標
```

```
p=a; //指標p設定為指標a的值
```

則

$*p$: a[0] 的資料，其值為 1

$p+1$: a[1] 的起始位址，即 $\&a[1]$

$p+2$: a[2] 的起始位址，即 $\&a[2]$

$*p+1$: 即 $(*p)+1$ ，其值為 $a[0]+1$

$*(p+1)$: a[1] 的資料，其值為 3

	a (靜態)
a[0]	1
a[1]	3
a[2]	5
a[3]	7
a[4]	9

練習(3)：執行並寫出結果

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a[]={1,3,5,7,9}, *p;

    for(p=a; p<=&a[4]; p++)
        printf("%d\n", *p);

    system("pause");
    return 0;
}
```

寫出執行結果

```
int main()
{
    int s [] = { 10 , 8 , 6 , 4 , 2 , 0 } , *q;
    q=s;
    printf ("%d\n" , *q) ;
    q++;
    printf ("%d\n" , *q) ;
    printf ("%d\n" , (*q)+3) ;
    printf ("%d\n" , * (q+2)) ;
    q+=3;
    printf ("%d\n" , *q) ;
}
```

兩數值交換

請更改程式使輸入的
兩數值交換
提示：傳遞指標



```
#include <stdio.h>
#include <stdlib.h>
void swap(int, int);
int main()
{
    int a, b;
    printf("輸入整數 a : ");
    scanf("%d", &a);
    printf("輸入整數 b : ");
    scanf("%d", &b);
    swap(a,b);
    printf("a=%d, b=%d\n", a, b);
    system("pause");
    return 0;
}

void swap(int x, int y)
{
    int temp;
    temp=x;
    x=y;
    y=temp;
    return;
}
```