

指標 && LINKEDLIST

資訊二甲

A thick yellow L-shaped graphic is positioned in the bottom right corner of the slide. It consists of a horizontal bar and a vertical bar that meet at a right angle.

指標

Pointer

變數三要素

變數位址

0X0012FF74

變數值

2

變數名稱

b

<https://www.goodfreephotos.com/astrophotography/milky-way-and-starry-night-sky.jpg.php>



Milky Way and
Starry Night Sky

<https://kopu.chat/2017/05/15/c%E8%AA%9E%E8%A8%80-%E8%B6%85%E5%A5%BD%E6%87%82%E7%9A%84%E6%8C%87%E6%A8%99%E5%BC%8C%E5%88%9D%E5%AD%B8%E8%80%85%E8%AB%8B%E9%80%B2%E5%BD%9E/>

指標 vs 指標變數

變數位址	變數值	變數名稱
0X0012FF74	2	b
0X0012FF90	0X0012FF74 (記憶體裡面)	pointer

指標 (Pointer) : 某變數的位址

指標變數 (Pointer Variable) : 用來存放指標的變數

宣告一個指標變數，和一般宣告變數一樣，
是跟記憶體要一個區域、存放這個變數的值。
只是這個變數的型別是指標。

宣告

- `type* name;` `/` `type * name;` `/` `type *name;` 皆可以 ~
- 一個 * 跟著一個變數，若：`int *a, b;` 只有 a 是指標變數，b 只是一般的整數變數。

- 透過 * 取值、&取址，例如：

```
int a, *b;
```

```
b = &a;
```

```
printf(“%d %d %p %p”, a, *b, &a, b); //印出數值再印出位址
```

MAN, I SUCK AT THIS GAME.
CAN YOU GIVE ME
A FEW POINTERS?

0x3A28213A
0x6339392C,
0x7363682E.

I HATE YOU.



練習一

輸入一串沒有特定數目的數字 (數字範圍皆於 0 ~ 1000 且不重複) ，
請使用 `void swap(int *a, int *b)` 函式 及 排序法 將數字從小排到大。

#Sample Input 1

1 5 9 4 2 3 8

#Sample Output 1

1 2 3 4 5 8 9

#Sample Input 1

493 273 95 0

#Sample Output 1

0 95 273 493

結構

Struct

結構

- 可以將彼此相關聯的變數包起來，成為一個新的資料型態 - 結構。
- 內部可以儲存多種資料型態，例如：

```
struct student{  
    char name[10];  
    int age;  
    char class;  
    struct student *next;  
}name1, name2, ...;
```

struct student	姓名	char[10]
	年齡	int
	班級	char
	下個人	struct student*

存取方式 (. Vs ->)

- 一般是使用 “.” 去提取結構裡的變數。
- 例如：name1.name, name1.age, name1.next, ...
- 但 “.” 的前方必須為 結構，不能是 指標。
- 若為指標，需使用 ->。

-> 範例

```
struct list {  
    int data;  
    struct list *next; // a pointer to struct list  
};
```

```
struct list listOne, listTwo, listThree;  
listOne.next = &listTwo;  
listTwo.next = &listThree; // 以下想要由listOne設定到listThree的data
```

```
listOne.next.next.data = 0; // 不合法, 因.左邊必須是struct,不可為pointer  
(*(*listOne.next).next).data = 0; // 這樣寫才對  
listOne.next->next->data = 0; // 這樣寫更漂亮
```

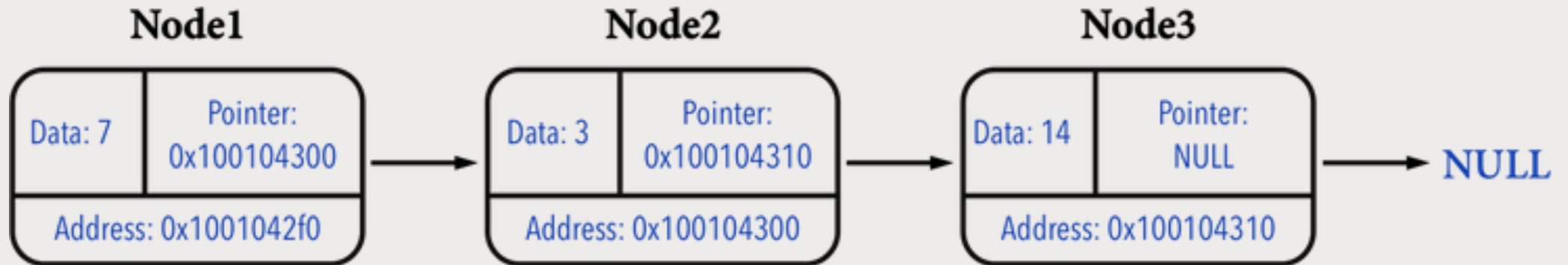
鏈結串列

Linked List



鏈結串列

- 使用 Node (節點) 做紀錄，並用 pointer 指向下一個 Node。
- 以 NULL 作為終點。

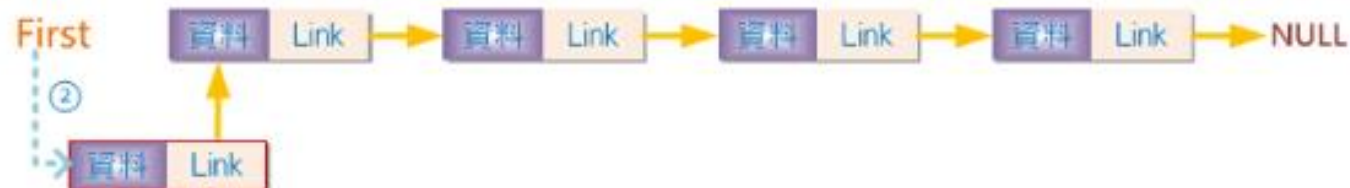


串接步驟

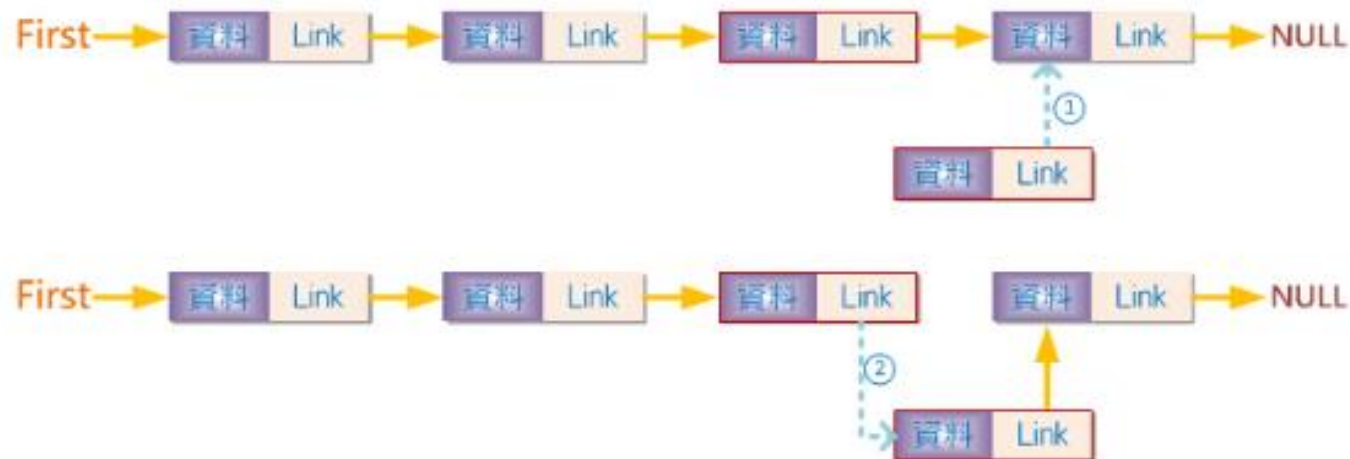
1. 建立一個 struct 當作 Node 的框架。
2. 建立新節點，並另一流動指標指向頭節點 (`ptr = head`)。
3. 建立另一個新節點，並把兩個節點串起來 (`ptr -> next = newnode`)。
4. 將流動指標指向剛剛建立的節點 (`ptr = ptr -> next`)。
5. 重複上面2和3的步驟。

新增

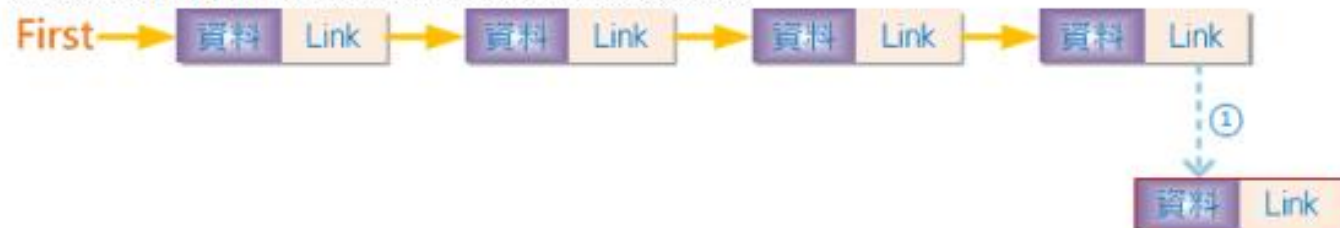
前端插入 (insertFront)：將新節點插入到串列最前端，並取代首節點



中間插入(insert)：將新節點插入到指定點後面

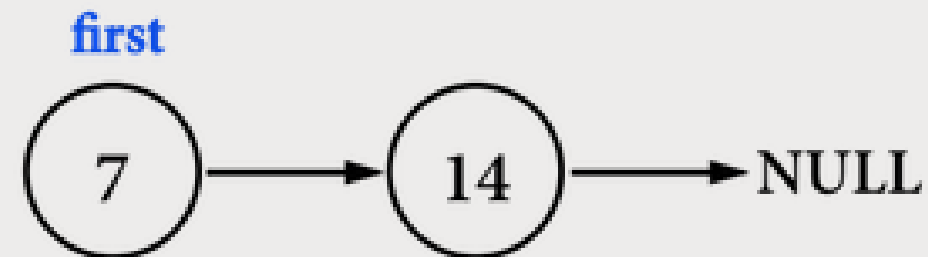
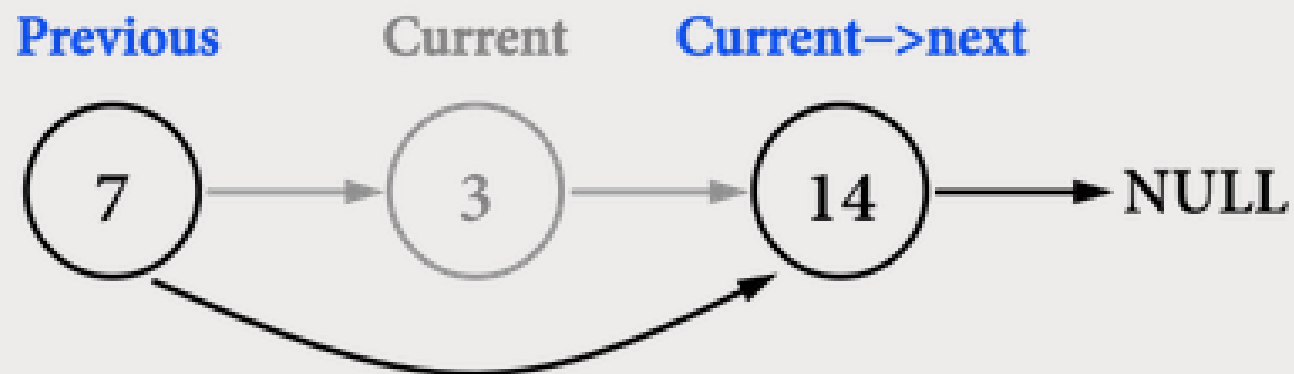
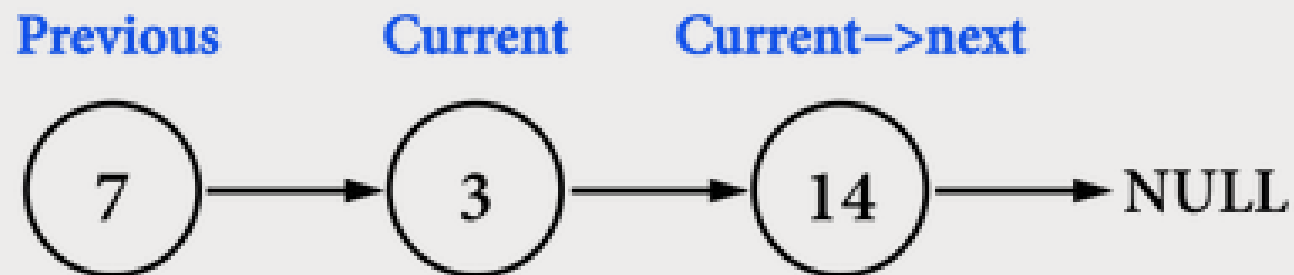


尾端插入(append)：將新節點插入到串列最尾端



删除

Case1-1: find x in list
eg: want to delete 3



練習二

有 N ($N < 100$) 個學生繞著圓圈而坐，按照順序排號 ($1 \sim N$) 。

從第一個人開始報數，凡是報到 3 的人退出圈子，

求最後留下來的的是原本編號幾的人呢？

請用 “鏈結串列” 實作。

#Sample Input 1

7

#Sample Output 1

4

A large yellow L-shaped graphic is positioned on the right side of the slide, consisting of a vertical bar and a horizontal bar meeting at a right angle.

END

快去寫練習吧