

## 鏈結串列自編教材（二）

- 將一鏈結串列第一個 node 所配置之記憶體釋放

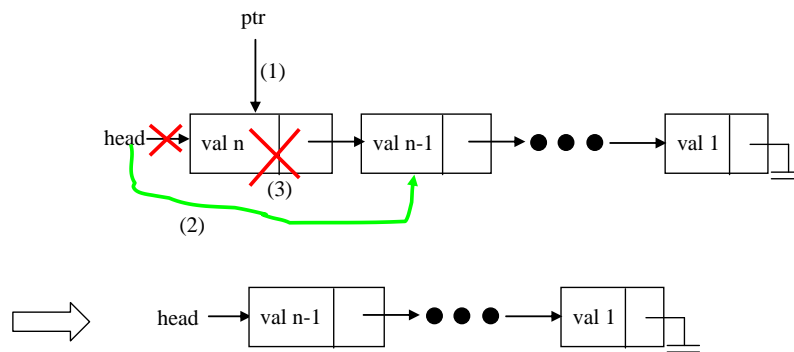


圖 2-1

```
-----  
ptr=head;//(1)  
head=head->next;//(2)  
free(ptr);//(3)  
-----
```

- 將一鏈結串列所配置之記憶體完全釋放(s3-12.c)

```
-----  
while (head!=NULL){  
    ptr=head;  
    head=head->next;  
    free(ptr);  
}  
-----
```

- 在鏈結串列尾端加入新的 node

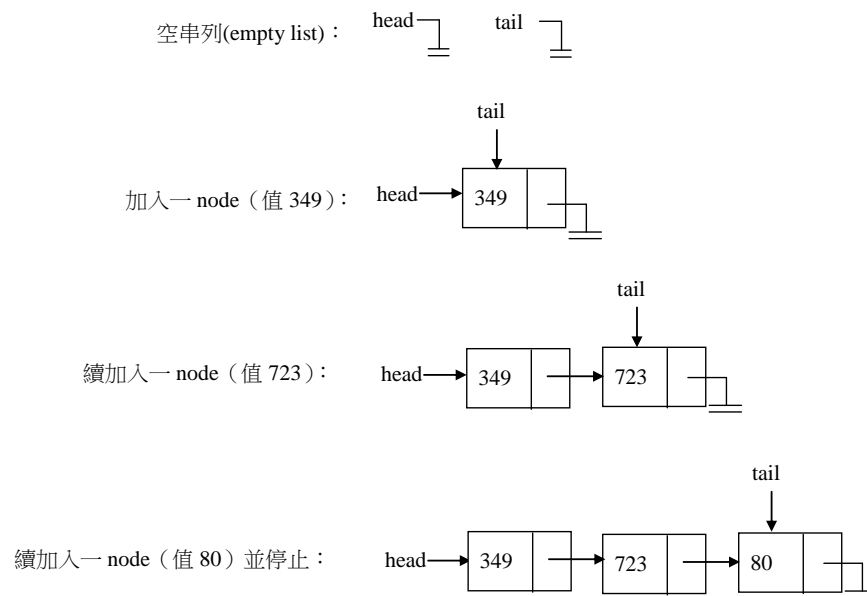


圖 2-2

- 空串列加入第 1 個 node

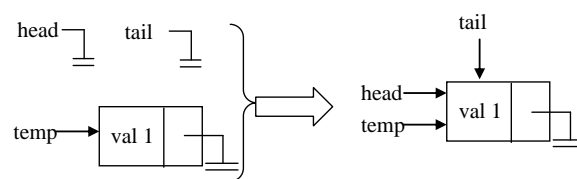


圖 2-3

```
-----  
if (head==NULL){  
    head=temp;  
    tail=temp;  
}  
-----
```

- 非空串列的尾端加入 1 個新 node

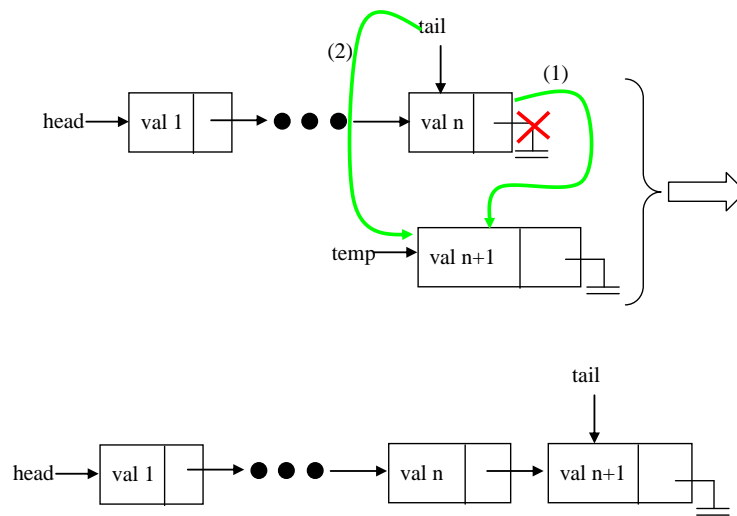


圖 2-4

-----  
tail->next=temp;//(1)

tail=temp;//(2)  
-----

- 每次以亂數產生一[0,1000]之整數值，若該值>100，則以同方式繼續產生下一亂數值，若該值≤100，則停止此產生值之程序。建一 linked list 儲存所有產生之值，且對每一新產生之值在 linked list 最尾端插入一新 node 以存放該值，並可重複列印所有產生的值，最後將該串列配置記憶體全部釋放。  
(s3-13.c)

```
-----  
struct ANODE *head=NULL, *tail=NULL, *temp;  
...  
//若加入新 node 之值>100，則繼續產生下一新 node  
do{  
    //產生一個新的 node  
    temp= generateanode();  
  
    if (head==NULL){//原來為空串列  
        head=temp;  
        tail=temp;  
    }  
    else{//將新產生的 node 加入串列的尾端  
        tail->next=temp;//(1)  
        tail=temp;//(2)  
    }  
} while (temp->val>100);  
-----
```