

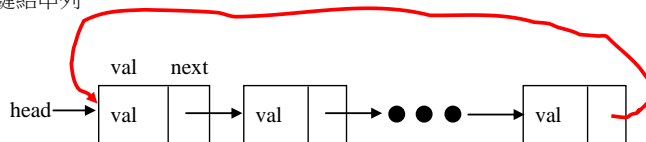
## 鏈結串列自編教材（五）

### ● 鏈結串列分類：

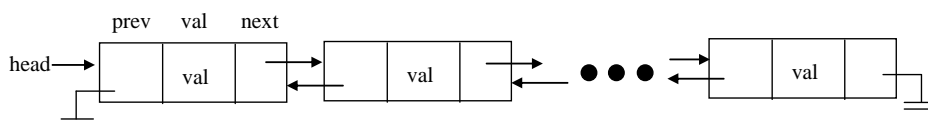
單向鏈結串列



環狀單向鏈結串列



雙向鏈結串列



環狀雙向鏈結串列

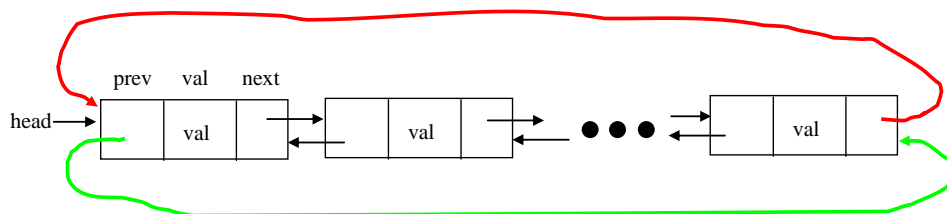


圖 5-1

- 以單向鏈結串列為基礎，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。所有產生之值必須全部記錄在環狀單向鏈結串列中(均在串列的最開頭新增一 node)，並可重複列印所有產生的值。

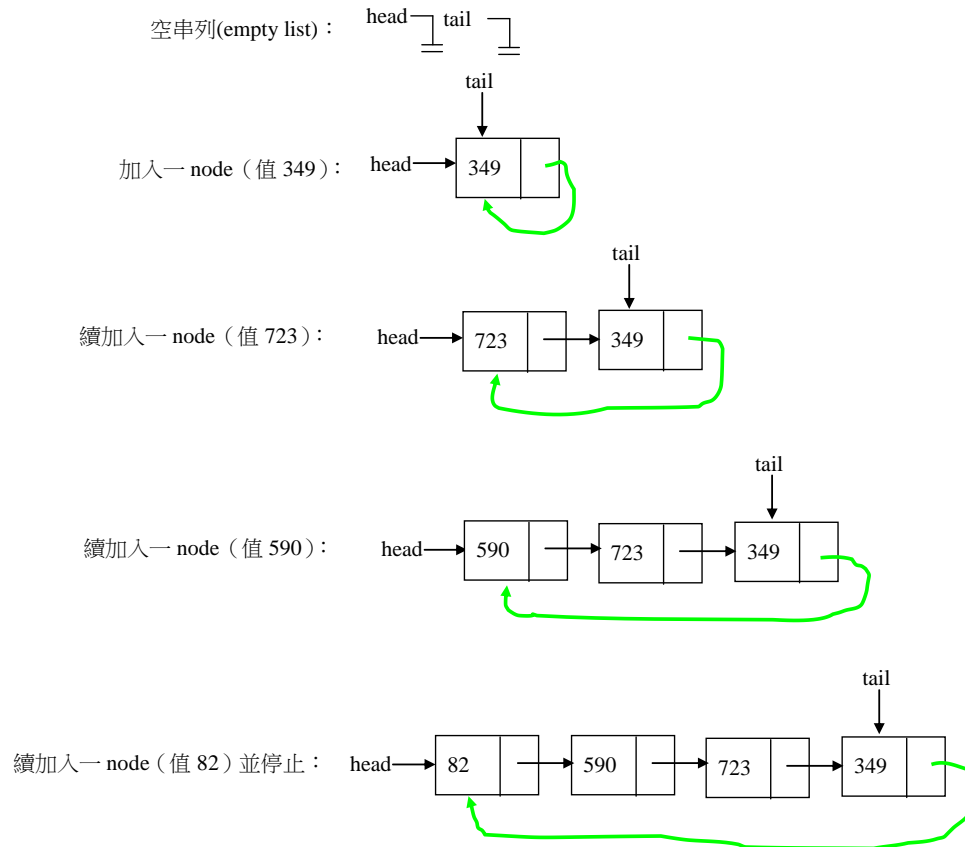


圖 5-2

- 空串列加入第 1 個 node

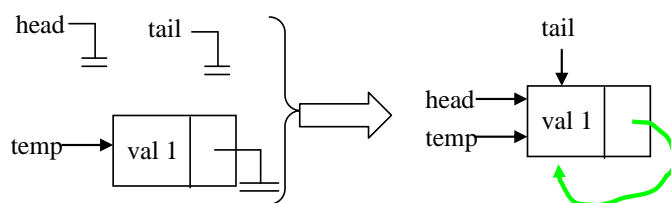


圖 5-3

```
temp= generateanode();
```

```
//指標 head 指向指標 temp，完成在空串列加入第 1 個 node
```

```
head=temp;
```

```
tail=temp;
```

```
head->next=head;//變成環狀串列
```

- 在非空串列開頭加入 1 個新 node

```
//注意順序性
```

```
temp->next=head;//(1)
```

```
head=temp;//(2)
```

```
tail->next=temp;//(3)
```

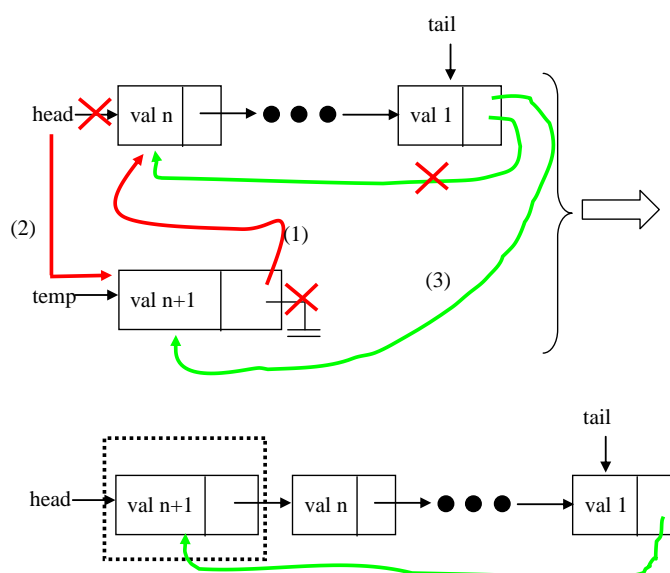


圖 5-4

- 列印環狀單向鏈結的內容

```
void printlist(){
    struct ANODE * ptr;

    if (head!=NULL){//不是空串列
        ptr=head;
        do{
            printf("%d\n",ptr->val);
            ptr=ptr->next;
        }while (ptr!=head);
    }//end of if
}
```

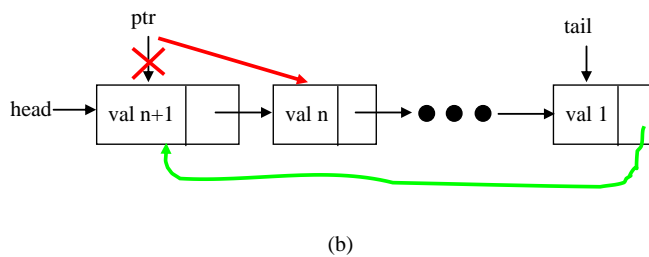
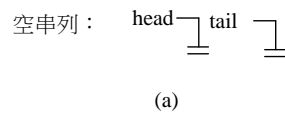


圖 5-5

- 以單向鏈結串列為基礎，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。所有產生之值必須全部記錄在環狀單向鏈結串列中(均在串列的最開頭新增一 node)，並可重複列印所有產生的值。(s3-16.c)

```
do{
//產生一個新的 node
temp= generateanode();

if (head==NULL){//原來為空串列
    //指標 head 指向指標 temp，完成在空串列加入第 1 個 node
    head=temp;
    tail=temp;
    head->next=head;//變成環狀串列
}
else{//將新產生的 node 加入串列適開頭
    temp->next=head;//(1)
    head=temp;//(2)
    tail->next=temp;//(3)
}
} while (temp->val>100);

printlist();
```

- 將環狀單向鏈結串列第一個 node 所配置之記憶體釋放(case a：至少有二個 nodes)

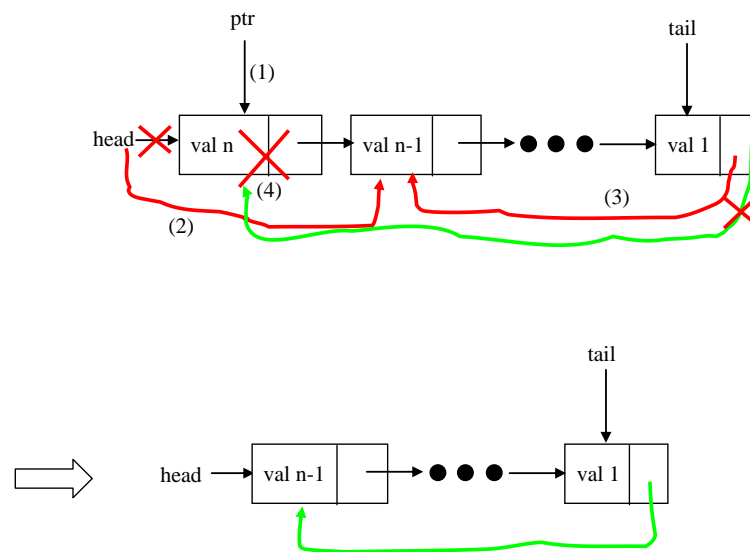


圖 5-6

```
-----  
ptr=head;//(1)  
head=head->next;//(2)  
tail->next=head;//(3)  
free(ptr);//(4)  
-----
```

- 將環狀單向鏈結串列第一個 node 所配置之記憶體釋放(case b：只有一個 node)

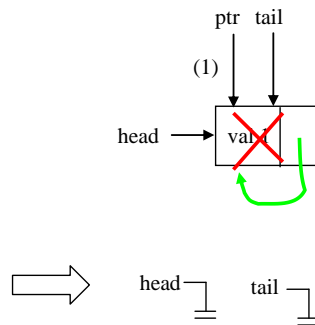


圖 5-7

```
ptr=head;//(1)
head=NULL;
tail=NULL;
free(ptr);
```

- 將前所產生的環狀單向鏈結串列所配置之記憶體完全釋放(s3-17.c)

```
if (head!=NULL){//串列非為空
    while (head!=tail){//串列含至少二個 node
        ptr=head;//(1)
        head=head->next;//(2)
        tail->next=head;//(3)
        free(ptr);//(4)
    }

    //串列只剩一個 node
    ptr=head;//(1)
    head=NULL;
    tail=NULL;
    free(ptr);
}
```