

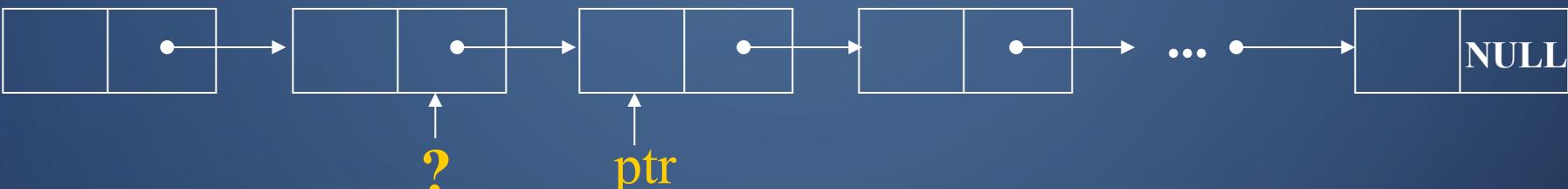


DATA STRUCTURE

07. DOUBLE LINKED LIST

DOUBLY LINKED LISTS (1/4)

- Singly linked lists 只能單向搜尋：沿著List方向搜尋時很快，但沒辦法反向



- 怎麼辦呢？加上一個往前的ptr！
- Doubly linked list: 3 fields
 - left link field(*llink*), data field(*item*), right link field(*rlink*).

```
typedef struct node *node_pointer;  
  
typedef struct node{  
    node_pointer llink;  
    element item;  
    node_pointer rlink;  
};
```

DOUBLY LINKED LISTS (2/4)

- 為了處理空list(沒有元素) , 我們加上head node
 - doubly linked circular with head node: (Figure 4.23)

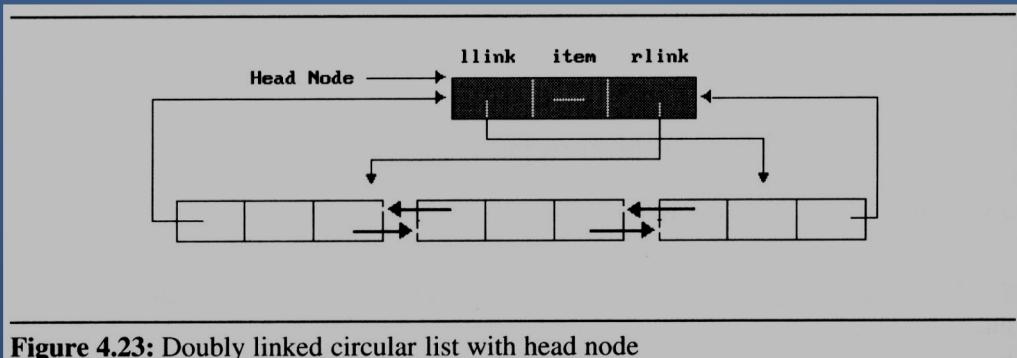


Figure 4.23: Doubly linked circular list with head node

- empty double linked circular list with head node (Figure 4.24)

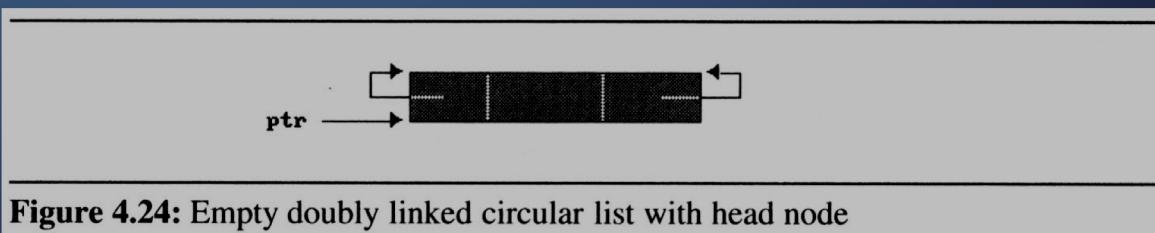


Figure 4.24: Empty doubly linked circular list with head node

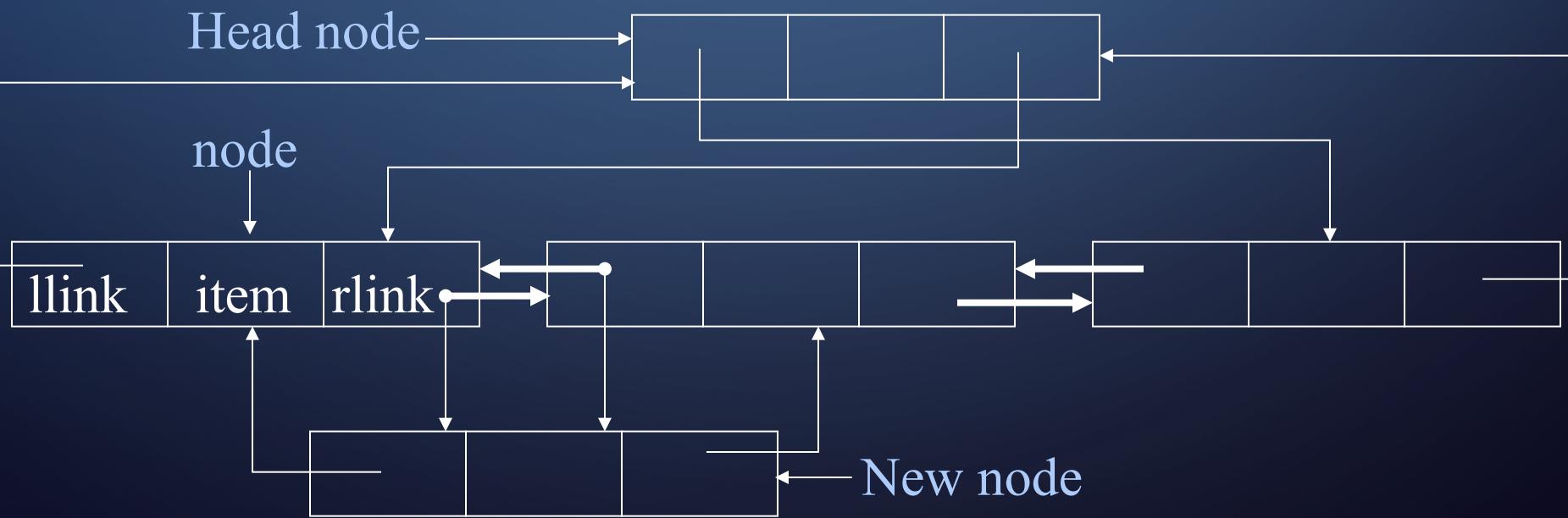
- suppose that *ptr* points to any node in a doubly linked list, then:
 - $\text{ptr} = \text{ptr} \rightarrow \text{llink} \rightarrow \text{rlink} = \text{ptr} \rightarrow \text{rlink} \rightarrow \text{llink}$

DOUBLY LINKED LISTS (3/4)

- Insert node

```
void dinsert(node_pointer node, node_pointer newnode)
{
    /* insert newnode to the right of node */
    newnode->llink = node;
    newnode->rlink = node->rlink;
    node->rlink->llink = newnode;
    node->rlink = newnode;
}
```

Program 4.28: Insertion into a doubly linked circular list

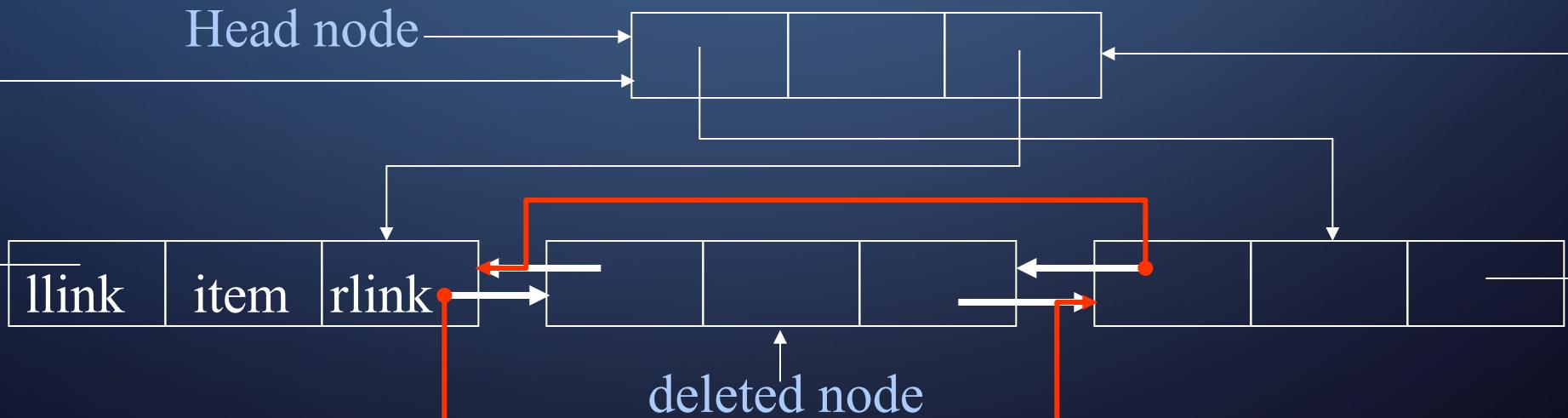


DOUBLY LINKED LISTS (4/4)

- Delete node

```
void ddelete(node_pointer node, node_pointer deleted)
{
    /* delete from the doubly linked list */
    if (node == deleted)
        printf("Deletion of head node not permitted.\n");
    else {
        deleted->llink->rlink = deleted->rlink;
        deleted->rlink->llink = deleted->llink;
        free(deleted);
    }
}
```

Program 4.29: Deletion from a doubly linked circular list



EXERCISE :

- 試著用Double Linked List做簡單的應用：迴文檢測
- AA, ABA, ABBBA... 等從左到右 和 從右到左相同的字串，稱為迴文
- 寫一個程式驗證輸入的字串是否是迴文

測資

- 首先有一個數字n($1 < n \leq 20$)，表示接下來有幾筆測資
- 每一筆測資一行，表示一個字串(由0-9跟英文字母包含大小寫)組成
- 請實作double linked list並利用pointer的機制檢查輸入是否是迴文
 - 單一個字也算迴文
 - 小心字數奇偶數

Input :

3
abba
ab
caac

Out :

Yes
No
Yes