Doubly Linked Lists 練習

- 使用雙向串列印出 1~6
- 在 3 後面插入 7
- 刪除 5

Doubly Linked Lists 練習

- 串列型態設定

```
typedef struct listNode* listPtr;
typedef struct listNode {
        int data;
        listPtr Rlink;
        listPtr Llink;
};
```

Doubly Linked Lists 練習

```
listPtr firstNode = NULL;
listPtr temp = NULL;
```

設定串列開頭

```
firstNode = (listPtr) malloc (sizeof(listPtr));
firstNode -> data = 1;
firstNode -> Llink = firstNode;
firstNode -> Rlink = firstNode;
temp = firstNode;
```

新增節點

```
int i;
for (i = 0; i < 5; i ++) {
        listPtr newNode = NULL;
        newNode = (listPtr) malloc (sizeof (listPtr));

        newNode -> data = i + 2;
        newNode -> Llink = temp;
        newNode -> Rlink = newNode;
        temp -> Rlink = newNode;

        temp = newNode;
}
```

連接串列頭尾

```
temp -> Rlink = firstNode;
firstNode -> Llink = temp;
```

VRMLAB
虛擬實境暨多媒體實驗室

Doubly Linked Lists 練習

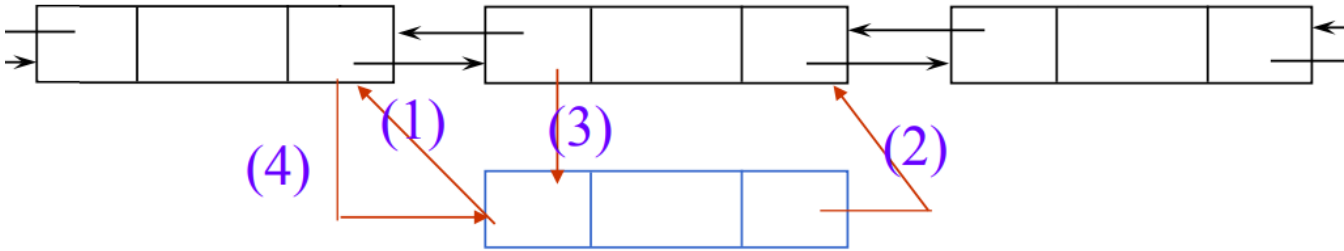
- 印出雙向串列

```
void print_list (listPtr first)
{
        listPtr temp = first;
        printf ("\n");

        do {
                printf ("\t%d", temp -> data);
                temp = temp -> Rlink;
        } while (temp != first);

        printf ("\n");
}
```

Doubly Linked Lists 練習

- 插入節點

```
void insert_node (listPtr node, listPtr newnode)
{
        newnode -> Llink = node;
        newnode -> Rlink = node -> Rlink;
        node -> Rlink -> Llink = newnode;
        node -> Rlink = newnode;
}
```



Doubly Linked Lists 練習

- 刪除節點

```
void delete_node (listPtr deletenode)
{
        deletenode -> Llink -> Rlink = deletenode -> Rlink;
        deletenode -> Rlink -> Llink = deletenode -> Llink;
}
```