

程式設計_bonus E94114073_張哲維_工科系 115

實現方法:

1. 新建節點(`createNode()`):使用 `malloc(sizeof())`分配一個內存，並將地址賦給 `newNode`，設置節點的值，並表示節點為最後一個。
2. 合併串列(`combineList()`):它會遍歷第一個串列，當找到最後一個節點後，將第二個串列連接到最後一個節點的 `next` 指針上。
3. 加入節點(`insertFirstNode()`):使用 `createNode(data)` 創建一個新節點，並將新節點的 `next` 指針指向原始串列的頭節點，即 `newNode->next = *head`。然後，將新節點設置為第一個位置，即 `*head = newNode`
4. 計算串列長度(`listLength()`):利用暫時的存取空間遍歷整個串列直到下一個節點為空值，即串列最後一個。
5. 反轉串列(`reverseList()`):使用了三個指針 `pre`、`cur` 和 `nextNode` 來逐個反轉節點。在循環中，首先保存當前節點的下一個節點為 `nextNode`，然後將當前節點的 `next` 指針指向前一個節點 `pre`，之後更新 `pre` 和 `cur` 指針，便利整個串列。最後將 `*head` 指針指向反轉後的串列的 `pre`。

心得:

實現函數功能裡的反轉函數是我覺得比較難的，我是參考了 [Linked List: 新增資料、刪除資料、反轉 \(alrightchiu.github.io\)](#)，了解了反轉功能的實現方法，而在程式中也一直重複練習函數的運用，以及動態記憶體的实际演練。此次的作業也使用了多次的指針去表示位址，所以在寫的過程中需要去小心的注意指標位址有無錯誤，否則將使程式無法執行抑或是錯誤。