PD-W16-A

LinkList

Q2:Insert&Delete
Ref:Ch17 P50~69

```c
#include<stdio.h>
#include<stdlib.h>

struct LinkedList{
    int data;
    struct LinkedList *next;
};
void Insert(struct LinkedList** head,int newData){
    struct LinkedList* newNode = (struct LinkedList*)malloc(sizeof(struct LinkedList));

    newNode->data = newData;
    newNode->next = *head;

    *head = newNode;
}

void Delete(struct LinkedList** head,int delData)
{
        struct LinkedList *cur , *prev = NULL;
        cur = *head;

        while(cur!=NULL)
        {
                if(cur->data==delData)
                        break;
                else
                {
                        prev = cur;
                        cur = cur->next;
                }
        }
        if(cur != NULL)        //有找到Data
        {
                if(prev != NULL)//位置不是head
                {
                        prev->next = cur->next;
                }
                else            //位置為head
                        *head =  cur->next;
        }
        free(cur);
}
void printLinkedList(struct LinkedList* head) {
    while (head != NULL) {
        printf("%d -> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}
int main()
{
    struct LinkedList *head=NULL;
    int i,del_num;

    for(i=0;i<10;i++)
        {
```

```
            Insert(&head,i);
        }
    printLinkedList(head);

    printf("Enter del_num:");
    scanf("%d",&del_num);
    Delete(&head,del_num);
    printLinkedList(head);

}
```

Q3.Reverse

tips:

you need **prev,current,next** pointer to LinkedList to implement Reverse

```
#include<stdio.h>
#include<stdlib.h>
struct LinkedList{
        int data;
        struct LinkedList *next;
};
void Insert(struct LinkedList** head,int newData){
    struct LinkedList* newNode = (struct LinkedList*)malloc(sizeof(struct LinkedList));

    newNode->data = newData;
    newNode->next = *head;

    *head = newNode;
}
void Reverse(struct LinkedList** head)
{
    struct LinkedList *prev,*current,*n;
    prev = NULL;
    current = *head;
    n = current->next;

    while(current!=NULL){
        n = current->next;
        current->next = prev;
        prev = current;
        current = n;
    }
    *head = prev;
}
void printLinkedList(struct LinkedList* head) {
    while (head != NULL) {
        printf("%d -> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}
int main()
{
    struct LinkedList *head = NULL;

    Insert(&head,3);
    Insert(&head,7);
    Insert(&head,10);


    printf("before:\n");
    printLinkedList(head);

    printf("after:\n");
    Reverse(&head);
    printLinkedList(head);
}
```