1、代码实现

#include<stdio.h>

#include<stdlib.h>

typedef int ElementType;

typedef struct Node \*PtrToNode;

struct Node {

ElementType Data;

PtrToNode Next;

};

typedef PtrToNode List;

List Read();

void Print(List L);

List Reverse(List L);

int main()

{

List L1, L2;

L1 = Read();

L2 = Reverse(L1);

Print(L1);

Print(L2);

return 0;

}

/\* 建立链表 \*/

List Read()

{

List head = NULL;

List current;

List prev = NULL;

int len = 0;

scanf("%d", &len);

if (len == 0) return NULL;

while (len--)

{

current = (List)malloc(sizeof(struct Node));

if (head == NULL)

head = current;

else

prev->Next = current;

current->Next = NULL;

scanf("%d", &current->Data);

prev = current;

}

return head;

}

void Print(List L)

{

List p = L;

List s = L;

List temp;

if (p == NULL)

printf("NULL");

else

printf("\n");

while (p!=NULL) {

printf("%d ", p->Data);

p = p->Next;

}

}

List Reverse( List L )

{

List Temp, Prev;

Prev = NULL;

while(L)

{

Temp = L->Next;

L->Next = Prev;

Prev = L;

L = Temp;

}

return Prev;

}

1. 实验总结

通过本次实验，我掌握了定义线性表的链式存储类型，加深了对链式存储结构的理解，进一步巩固和理解了单链表的基本操作，建立、查找、插入和删除等等。也体会到了链表在插入、删除操作中的优点及链表中添加头结点的优点，便于首元结点的处理以及空表和非空表的统一处理。