---

layout: post

tags: C 代码练习 单链表

title: C语言练习（11）--单链表就地逆置等

---

使用工具：DEV C++

涉及算法：

1. 倒序输出单链表

1. 删除最小值节点（假设唯一）

3. 就地逆置，辅助空间为O(1)，时间复杂度O(n^2)

```c

#include <stdio.h>

#include <stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}Node, \*LinkList;

//头插法构建单链表

void creatLinkList(LinkList L) {

Node\* s;

int x;

scanf("%d", &x);

while(x!=9999) {

s = (Node\*)malloc(sizeof(Node));

s->data = x;

s->next = L->next;

L->next = s;

L->data ++;

scanf("%d", &x);

}

printf("\nlength:%d", L->data);

}

//顺序输出单链表

void printLinkList(LinkList L) {

printf("\n顺序输出单链表：");

while(L->next!=NULL) {

printf("%d ", L->next->data);

L = L->next;

}

}

//倒序输出单链表

void reversePrintLinkList(LinkList L) {

int data[L->data];

int i = 0;

while(L->next!=NULL) {

data[i++] = L->next->data;

L = L->next;

}

printf("\n倒序输出单链表：");

while(i > 0) {

printf("%d ", data[--i]);

}

}

//删除最小值节点（假设唯一）

void deletMin(LinkList L) {

Node \*p = L;

Node\* s = (Node\*)malloc(sizeof(Node));

if(L->next == NULL) return;

s = L;

while(L->next!=NULL) {

if(L->next->data < s->next->data) {

s = L;

}

L = L->next;

}

printf("被删除的数：%d",s->next->data);

//删除最小值点的过程

Node \*t = s->next;

s->next = t->next;

free(t);

p->data--;

}

//就地逆置，辅助空间为O(1)，时间复杂度O(n^2)

void reverseLinkList(LinkList L) {

int i, j;

Node\* s = L;

int length = L->data;

L = L->next;

for(i = 1; i < length; i++) {

L = s->next;

for(j = 0; j < length-i ; j++) {

change(L);

L = L->next;

}

}

}

void change(Node\* p) {

int temp;

temp = p->data;

p->data = p->next->data;

p->next->data = temp;

}

int main(int argc, char \*argv[]) {

LinkList L;

L = (LinkList)malloc(sizeof(Node));

L->next = NULL;

L->data = 0;

creatLinkList(L);

printLinkList(L);

reversePrintLinkList(L);

deletMin(L);

printLinkList(L);

printf("\n删除后length：%d",L->data);

reverseLinkList(L);

printLinkList(L);

return 0;

}

```

结果图：

![](/assets/img/2016-07-28-C11/1.png)

