

• 单链表求逆

新建一个单链表，改变其为逆序。即给定一个M个节点的链表，将其链接顺序颠倒。当M=9时
Head->1->2->3->4.....->9->NULL

逆序为

Head->9->8->7.....->3->2->1->NULL

```
typedef struct tagNODE{
    int data;
    struct tagNODE* next;
} STNODE;

int main()
{
    int M=9;
    STNODE* Head;
    Head=(STNODE*)malloc(sizeof(STNODE));
    STNODE* Temp=Head;
    for(int i=1;i<=M;i++) //尾插法建立单链表，按顺序赋值
    {
        STNODE* newNODE=(STNODE*)malloc(sizeof(STNODE));
        newNODE->data=i;
        Temp->next=newNODE;
        newNODE->next=NULL;
        Temp=newNODE;
    }
    //打印原来的链表
    for(STNODE* temp=Head->next;temp!=NULL;temp=temp->next)
        printf("%d->", temp->data);
    printf("\n");
    //进行反转。
    //要进行反转大概要有三个指针
    STNODE* Reverse=NULL; //已经反转的链表
    STNODE* Visit=Head->next; //当前需要反转的节点
    STNODE* Last=Head->next; //还没有反转的链表

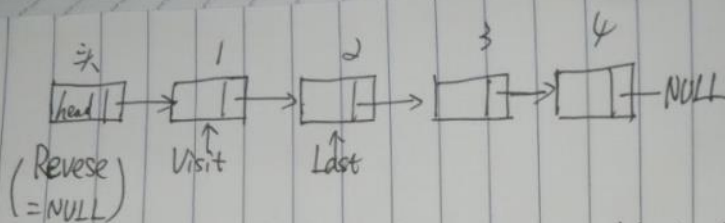
    while(Visit!=NULL) //当前需要反转的为空 那么结束
    {
        Last=Last->next; //位反转节点后移一位

        Visit->next=Reverse; //将当前结点插入到已经反转的链表头部
        Reverse=Visit; //已经反转的头部变成当前结点
        Visit=Last; //当前结点更换为还未反转的节点，循环继续反转
    }
    STNODE* newHead=(STNODE *)malloc(sizeof(STNODE)); //申请一个新的头指针指向反转的
链表
    newHead->next=Reverse;

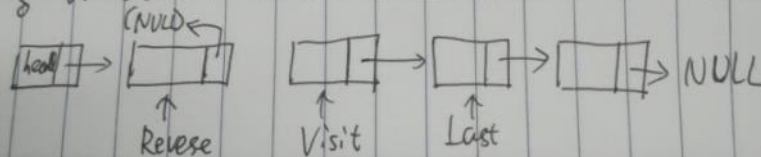
    //打印反转的链表
    for(STNODE* temp=newHead->next;temp!=NULL;temp=temp->next)
        printf("%d->", temp->data);
}
```

D:\visual_studio_Projects\OpencvO

1->2->3->4->5->6->7->8->9->
9->8->7->6->5->4->3->2->1->



Visit → next = Reverse ; Reverse = Visit ; Visit = Last ; 即



看到结点 1 被成功反转. 接着循环执行. 就可以反转整张链表. 这里最主要是要维护好这三个指针. 因为在进行反转时. 必须将链表分成三个部分. 1) 已经反转完毕的 Reverse 指向的链表. 2) 正要反转的结点. 3) 还未反转的链表 Last.