#include <stdio.h>

#include <conio.h>

#include<malloc.h>//#include "alloc.h"

typedef struct /\* 使用typedef定义类型 \*/

{

char x;

struct node \* next;

} node;

node \* input()

{

node \*p1,\*p2,\*h=NULL;

char ch;

while(1)

{

ch=getche();

if(ch=='\r') break;

p1=(node \*)malloc(sizeof(node));

p1->x=ch;

if(h==NULL) h=p1;

else p2->next=p1;

p2=p1;

}

p1->next=NULL;

return h;

}

void display(node \*p)

{

printf("\n");

while(p!=NULL)

{

putchar(p->x);

p=p->next;

}

}

node \* reverse(node \* p)

{

node \* p1=NULL,\*p2;

while(p!=NULL)

{

p2=(node \*) malloc(sizeof(node));

if(p1==NULL) p2->next=NULL; /\* 逆转之后，原链表的头结点就是新链表的尾结点 \*/

else p2->next=p1; /\* 如果不是第一个结点，则本次产生的新结点是上次结点的前一个 \*/

p1=p2;

p2->x=p->x;

p=p->next;

}

return p1; /\*原链表的最后一个结点是新链表的头结点 \*/

}

int main(void)

{

node \* head,\* h2;

head=input();

display(head);

h2=reverse(head);

display(h2);

getch();

return 0;

}

**借助辅助空间**

建立临时的新链表，将新节点指向其前驱结点实现逆转，结合注释可以得到思路。